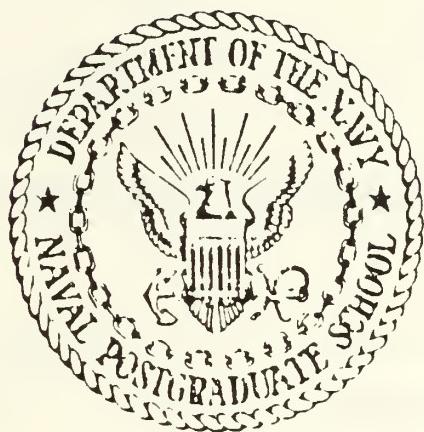




DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943-8002

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

AUTOMATED POLE PLACEMENT ALGORITHM

FOR MULTIVARIABLE OPTIMAL CONTROL SYNTHESIS

by

Chow, Wah Keh

September 1985

Thesis Advisor:

D. J. Collins

Approved for public release; distribution unlimited.

T226057

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Automated Pole Placement Algorithm for Multivariable Optimal Control Synthesis		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1985
7. AUTHOR(s) Chow, Wah Keh		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE September 1985
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 170
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Linear Quadratic Control, Pole Placement, Pole Assignment, Multivariable System, Multi-input Multi-output Control, Multivariable System, Robustness, Weighting Matrix.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This work addresses the application of numerical optimization technique to the pole-placement problem in multivariable optimal control. An algorithm is developed to select a set of weighting matrix element such that the conventional transient response criteria are satisfied. General properties of the optimal system in terms of stability, robustness and relative weights between state and control		

variables were explored by applying the method to the design of two multivariable systems. Results indicated that this method provides good insight to the problem for the designer and is therefore a useful tool in multivariable control synthesis.

Approved for public release; distribution is unlimited.

Automated Pole Placement Algorithm
for Multivariable Optimal Control Synthesis

by

B.A. (Hons),^{Chow, Wah Keh} University of Oxford, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1985

Thesis
4881

ABSTRACT

This work addresses the application of numerical optimization technique to the pole-placement problem in multivariable optimal control. An algorithm is developed to select a set of weighting matrix element such that the conventional transient response criteria are satisfied.

General properties of the optimal system in terms of stability, robustness and relative weights between state and control variables were explored by applying the method to the design of two multivariable systems. Results indicated that this method provides good insight to the problem for the designer and is therefore a useful tool in multivariable control synthesis.

TABLE OF CONTENTS

I.	INTRODUCTION	11
	A. BACKGROUND	11
	B. OVERVIEW	13
II.	MULTIVARIABLE LINEAR QUADRATIC CONTROL	14
	A. LINEAR QUADRATIC SYSTEM	14
	B. POLE ASSIGNMENT PROBLEM	17
	C. WEIGHTING MATRICES AND SYSTEM PERFORMANCES . .	19
III.	ROBUSTNESS THEORY AND LINEAR QUADRATIC CONTROL . .	24
	A. ROBUSTNESS CONCEPTS	24
	B. ROBUSTNESS IN LQ SYSTEM	30
IV.	DESIGN PROCEDURE	32
	A. GENERAL	32
	B. APPROACHES	32
	C. POLE PLACEMENT ALGORITHMS	34
	1. System with Real and Distinct Eigenvalues	34
	2. System with Complex Eigenvalues	37
	3. System With Repeated Eigenvalues	38
	D. COMPUTER PROGRAM DESCRIPTION	39
	E. DESIGN PHILOSOPHY	40
V.	DESIGN EXAMPLES	44
	A. INTRODUCTORY 2X2 PROBLEM	44
	B. A HELICOPTER DESIGN PROBLEM	56
	C. DISCUSSION AND CONCLUDING REMARKS	82
VI.	CONCLUSIONS	94

APPENDIX A: NON-DIAGONAL R DESIGNS	96
APPENDIX B: EXAMPLE DESIGN RUN OUTPUTS	106
APPENDIX C: COMPUTER PROGRAM LISTINGS	116
LIST OF REFERENCES	168
INITIAL DISTRIBUTION LIST	170

LIST OF TABLES

I	PERTURBED EIGENVALUE FOR 2X2 MODEL	48
II	PERTURBED EIGENVALUE FOR LQ DESIGN	53
III	ALPHATECH DESIGNS (1, 2, AND 3)	65
IV	RESULTS FROM POLE PLACEMENT SEQUENCE(LQ-A)	67
V	RESULTS FROM POLE PLACEMENT SEQUENCE(LQ-B)	68
VI	CLOSED-LOOP EIGENVECTORS	69
VII	RESULTS FROM POLE PLACEMENT SEQUENCE (LQ-C)	99
VIII	RESULTS FROM POLE PLACEMENT SEQUENCE (LQ-D) . . .	100

LIST OF FIGURES

2.1	State Feedback System-Time and Frequency Domain . . .	16
2.2	Selecting Relative Weighting Parameter ρ	20
2.3	Single Input Asymptotic Root Loci	22
2.4	Multiple Input Asymptotic Root Loci	23
3.1	The Perturbed SISO System	25
3.2	The Nyquist Plot - Gain and Phase Margin	26
3.3	The Minimum Guaranteed GM and PM	27
3.4	The Perturbed MIMO System	28
3.5	SISO LQ Nyquist	30
4.1	Computer Aided Design Package Organization	41
4.2	LQ Eigenvalue Assignment Philosophy	43
5.1	A Simple 2X2 MIMO Model	45
5.2	Unity Feedback for the 2X2 Model	47
5.3	Loop Transfer Matrix - $F = I$	49
5.4	Multivariable Nyquist Plot	50
5.5	Singular Value Plots - Unity Feedback	51
5.6	Comparison-Singular Value Plots (2x2 model)	54
5.7	Closed-Loop Pole-Zero Plots (Unity Gain FB)	56
5.8	Closed-Loop Pole-Zero Plots (LQ Design)	57
5.9	Open-Loop Bode Plots Channel 1-1	58
5.10	Open-Loop Bode Plots Channel 2-1	59
5.11	Open-Loop Bode Plots Channel 2-2	60
5.12	State Variables and Body Axes for CH-47	62
5.13	Open Loop Time Response $\phi(0) = 0.1$ rad	64
5.14	Closed-Loop Time Response Plot (AlphaTech 1)	70
5.15	Closed-Loop Time Response Plot (LQ-A)	71
5.16	Closed-Loop Time Response Plot (LQ-B)	72
5.17	Closed-Loop Pole-Zero Plots (AlphaTech 1)	74

5.18	Closed-Loop Pole-Zero Plots (LQ-A)	75
5.19	Closed-Loop Pole-Zero Plots (LQ-B)	76
5.20	Bode Plots Comparison- Input 1-1	78
5.21	Bode Plots Comparison - Input 1-2	79
5.22	Bode Plots Comparison - Input 2-1	80
5.23	Bode Plots Comparison - Input 2-2	81
5.24	Closed-Loop Bode Plot- AlphaTech 1-1	83
5.25	Closed-Loop Bode Plot- AlphaTech 1-3	84
5.26	Closed-Loop Bode Plot- AlphaTech 1-4	85
5.27	Closed-Loop Bode Plot- LQ-A 1-1	86
5.28	Closed-Loop Bode Plot- LQ-A 1-3	87
5.29	Closed-Loop Bode Plot- LQ-A 1-4	88
5.30	Closed-Loop Bode Plot- LQ-B 1-1	89
5.31	Closed-Loop Bode Plot- LQ-B 1-3	90
5.32	Closed-Loop Bode Plot- LQ-B 1-4	91
A.1	Singular Value Plots - Comparison	101
A.2	Bode Plots Comparison- Input 1-1	102
A.3	Bode Plots Comparison - Input 1-2	103
A.4	Bode Plots Comparison - Input 2-1	104
A.5	Bode Plots Comparison - Input 2-2	105

ACKNOWLEDGMENTS

I wish to gratefully acknowledge my thesis advisor, Prof. D. J. Collins, for suggesting the basis of this thesis work, and for his invaluable advice and guidance during the course of the thesis.

I would also like to express my appreciation to Prof. J. V. Sanders and R. D. Zucker for their support during my study at NPS, and to all others who have helped me directly or indirectly to successfully complete this thesis.

I. INTRODUCTION

A. BACKGROUND

For more than two decades, the need to solve control problems in aerospace application has been the primary driving force behind the modern control theory development. Problems in manoeuvring, guidance and tracking of aircrafts and space vehicles have motivated the development of various control design and synthesis methods. One of the methods, the so-called Linear Quadratic Control or LQ-control [Refs. 1,2] has been widely used and is treated extensively in the control literatures. Unlike most classical methods where the design are based on conventional time response criteria, the LQ theory treats the problem of designing controllers as that of minimizing a quadratic cost function of states and control inputs. The design problems become that of selecting suitable weighting matrices in the performance index. Two questions naturally arise from this method. First, how does one select the weighting matrices and second, once a set weighting matrices is selected, how does one know that it is a good design. There are generally two approaches to the first problem; the obvious one is to rely on physical arguments and a certain amount of trial and error [Refs. 3,4]. Unfortunately, such formulation can be obtained in only a few cases. Reference 4 provides a few guidelines that can be employed. The second approach is to avoid the physical aspect of the performance index but instead try to relate the weighting matrices with some other performance specification. For example, Tyler and Tuteur [Ref. 5] expressed the characteristic polynomial as an explicit function of the weighting matrices for single input

single output (SISO) system with diagonal weighting elements. Root-Locus type of procedure were used to show how the variation in weighting elements affect the eigenvalues of the closed-loop system. Similar relations were explored in [Refs. 6,7] in which more general expressions were obtained. Their uses were restricted to single input case due to difficulty in handling polynomial matrices. Solheim [Ref. 8] later developed a sequential design procedure based on diagonalized (decoupled) system. More recent results of eigenvalue placement in optimal control problem are presented in [Refs. 9,10,11,12].

The second issue of LQ design, i.e., whether a good design has been obtained once the performance index has been fixed, is related to the multivariable nature of system. Unlike the single input case where the closed-loop eigenvalues uniquely define the feedback gain and hence the weighting matrices, the MIMO structure provides additional dimension which allows further tradeoff for properties other than the closed-loop eigenvalue location. An example is the gain and phase margin, or in MIMO case, the so-called robustness criteria. Robust-control has been the subject of extensive researches in recent years and results relating to optimal control can be found in [Refs. 13,14,15].

It becomes evident from the above discussion that LQ control design and synthesis are not just a matter of specifying performance index. Optimal in the sense of satisfying performance index and perhaps pole location does not necessary means that a good design has been obtained. Other criteria like disturbance rejection, robustness and sensitivity need to be considered and incorporated in the design procedure. This is the motivation behind our present research which in turn leads to the development of the synthesis package. An overview of the thesis is described in the next section.

B. OVERVIEW

In this section an overview of the thesis is given. A background of multivariable optimal control theory in terms of its structure, frequency domain characteristic and asymptotic properties is first discussed together with derivation of some useful relationships in Chapter 2. General robustness concepts and its application in Linear Quadratic (LQ) Control is presented in Chapter 3. In Chapter 4, a computer aided design package for pole-placement synthesis based on numerical optimization technique is presented. This package provides a useful computational tool to support the material in the remaining chapter. A step-by-step pole-placement synthesis procedure is also presented to illustrate how the package can be used to design optimal control system that meet time response criteria and other properties. The use of the pole-placement synthesis package in two actual design problems is demonstrated in Chapter 5. Results in terms of frequency and time response properties ,robustness (singular value decomposition) are compared to those obtained by other design methods. Program listings and an example of a design session are included in the Appendix.

II. MULTIVARIABLE LINEAR QUADRATIC CONTROL

In this Chapter a brief review of Multivariable Linear Quadratic Control theory is presented. Structure of LQ system is first given, general stability properties are then discussed. In section two the general pole assignment problem is formulated for the MIMO state feedback system. Interpretation of the state and control input weighting matrices and their effect on the closed-loop time response behaviors and the asymptotic properties are presented.

A. LINEAR QUADRATIC SYSTEM

Consider the following linear time-invariant state-space system given by the equations:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (\text{eqn 2.1})$$

$$y(t) = Cx(t) \quad (\text{eqn 2.2})$$

where $x(t)$ is the n -dimensional state vector, $u(t)$ is the m -dimensional control vector and $y(t)$ is the p -dimensional output vector. A , B and C are real constant matrices of dimension $n \times n$, $n \times m$ and $p \times n$ respectively. Assuming that they form a controllable pair (A, B) and an observable pair (A, C) , the optimal feedback control law is obtained by minimizing the following quadratic performance index.

$$J = \int_0^{\infty} (x^T Q x + \rho u^T R u) dt \quad (\text{eqn 2.3})$$

where R is positive definite ($R > 0$) for bounded input and ρ is a scalar. Q is a semi-positive definite matrix ($Q \geq 0$).

When both Q and R are diagonal matrices, ρ defines the relative weight between the state and control weighting matrices in the performance index. The quadratic form $x^T Qx$ and $u^T Ru$ provide a weighted measure of the magnitude of the states and control vector respectively. The steady state control law that minimizes J is given by,

$$u(t) = -Fx(t) \quad (\text{eqn 2.4})$$

where F is the feedback gain matrix which is given by

$$F = -R^{-1}B^T P \quad (\text{eqn 2.5})$$

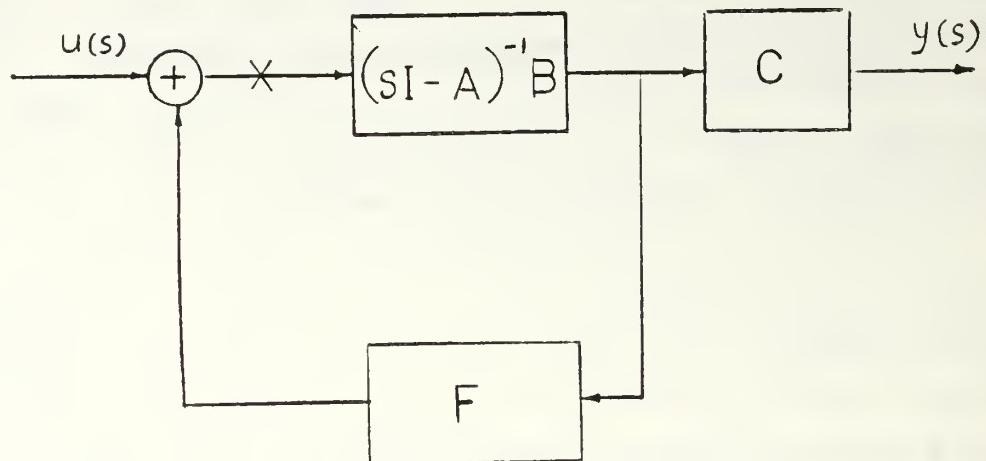
The positive definite matrix P is given by the solution to the steady state Riccati equation.

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \quad (\text{eqn 2.6})$$

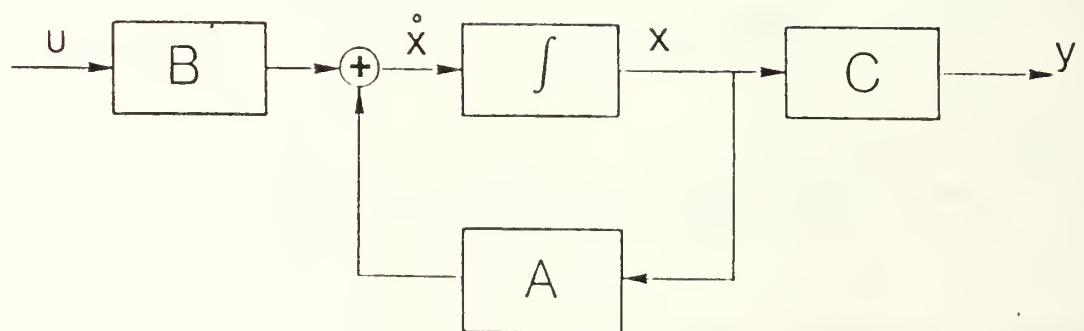
Equation 2.5 and 2.6 are well-known results in optimal control theory that yield the optimal closed-loop system

$$\dot{x}(t) = [A + BF]x(t) \quad (\text{eqn 2.7})$$

whose closed-loop poles are given by the eigenvalues of the matrix $[A+BF]$. The LQ system given by equations 2.1 through 2.6 is closed-loop stable and can be represented by the feedback configuration in both the time and frequency domain as shown in Figure 2.1 .



Frequency Domain



Time Domain

Figure 2.1 State Feedback System-Time and Frequency Domain.

In Figure 2.1, if the loop is broken at the input (X) as shown, the loop transfer function is given by

$$G(s) = F(sI - A)^{-1}B \quad (\text{eqn 2.8})$$

The matrix $[I + G(s)]$ is called the Return Difference Matrix and will be shown to have some important feedback properties in the following section.

B. POLE ASSIGNMENT PROBLEM

In the most general form, the state feedback pole assignment problem in control system design can be formulated precisely as follows:

"Given real matrices (A, B) of order (nxn, nxm) respectively and a set of n complex numbers $(\lambda_1, \lambda_2, \dots, \lambda_n)$ closed under complex conjugation. Find a real $m \times n$ matrix F such that the eigenvalues of $[A+BF]$ are λ_i , $i = 1, 2, 3, \dots, n$."

In general, the closed-loop eigenvalues of $(A+BF)$ can be arbitrary located in the complex plane, with the only restriction that complex characteristic eigenvalues must occur in complex conjugate pair. In other words, if the matrices (A, B) are a completely controllable pair, the stability of the system can always be improved by state feedback. If the (A, B) pair is not completely controllable, then the system is required to be 'stabilizable' meaning that in its controllability canonical form given by equation 2.9 below, the matrix A_{22} is asymptotically stable or any unstable subspace of equation 2.9 is also in the controllable subspace.

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

(eqn 2.9)

The solution to the pole assignment problem in the single input ($m=1$) case, when it exists, can be shown to be unique. In the multiple input case ($1 < m < n$), the solution of the so-called inverse eigenvalue problem is, in general, undetermined with many degrees of freedom. Additional conditions must be supplied in order to eliminate the extra degrees of freedom. This has been an area of active research in recent years and a number of approaches have been developed to relate the extra degrees of freedom with properties such as system eigenvectors, transmission zeros and robustness [Refs. 16,17]. In this work, it is shown that Linear Quadratic formulation incorporating equation 2.3 will partly remove the uncertainty that exists in the multiple inputs case. It will be shown that useful properties like robustness are guaranteed. It is also shown that the LQ type of pole placement formulation, when combined with eigenvectors assignment type of formulation, will produce some very useful design and synthesis procedures. In the present work, however, only the LQ eigenvalue placement problem is addressed; the problem of combined eigenvalue and eigenvector assignment is briefly described in Chapter 5.

C. WEIGHTING MATRICES AND SYSTEM PERFORMANCES

This section briefly reviews the effect of weighting matrices on system performance. The physical aspects of the weighting matrix for both single and multiple input systems are presented first together with some discussion on the asymptotic behavior of the LQ system.

It was shown in the last section that under complete controllable conditions, the time-invariant linear system can be stabilized by a linear feedback control law. For the regulator-type problem where the aim is to bring the system from an arbitrary initial state to the zero state, the closed loop poles can be chosen far to the left on the complex plane. Convergence to the zero state is fast but the large input required may not be practical. This naturally leads to an optimization problem where the trade off is between speed of convergence to zero and the magnitude of the input. This is reflected in the two quadratic terms in the performance index. The quantity $x^T Q x$ in the first term of the performance index is a measure of the extent in which the state at time t deviates from the zero state. The matrix Q determines how much weight is attached to each of the component of the state. The integral $\int (x^T Q x) dt$ is a criteria for the cumulative deviation of $x(t)$ from the zero state during the interval.

The problem of large control input is resolved by incorporating the second quadratic term, $\int (u^T R u) dt$. Larger value in the element of the control weighting matrix R will result in smaller input. It will be shown later that one can manipulate R to achieve some secondary design objectives. The remaining parameter ρ , that need to be specified, accounts for the relative weighting between the state and the control inputs. Selecting optimal value for ρ depends on the particular problem and the design requirement. As ρ

decreases, the integrated square regulating error decreases but the integrated square input increases. Very often, the optimal control problem is solved for many different values of ρ . A graph similar to that shown in figure 2.2 is obtained where the integrated square regulating error is plotted versus the integrated square input. An appropriate value of ρ is then selected to give sufficiently small regulator error without excessively large input.

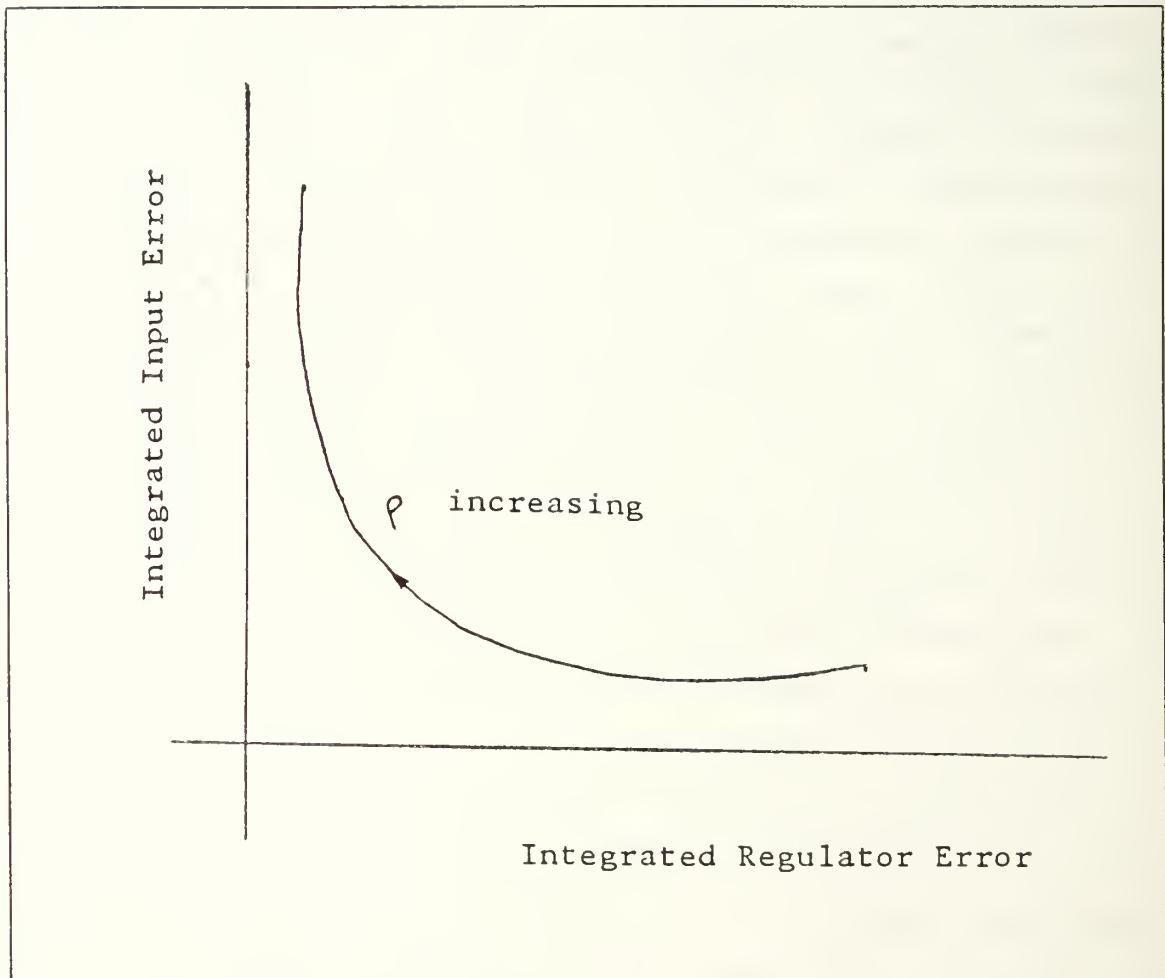


Figure 2.2 Selecting Relative Weighting Parameter ρ .

The special case where ρ approaches zero, the so-called asymptotic properties, has been shown to provide good insight for LQ control system design. Some of the results from [Refs. 4,18] are summarized below.

As ρ decreases to zero for the system given by equations 2.1 through 2.6, some (say q) of the closed-loop poles go to infinity while other ($n-q$) stay finite. Those remaining finite approach the left half plane zero of $\det[B^T(-sI-A^T)^{-1}Q(sI-A)^{-1}B]$. If m is the dimension of the input vector, then at least m closed-loop poles approaches infinity. All closed-loop poles that go to infinity do so by grouping into several Butterworth patterns of different order and radii. For ρ approaching infinity, the closed-loop poles approach the mirror image of the plant open-loop poles. To illustrate the asymptotic concept, two examples from [Ref. 4] are given below.

An example of the asymptotic properties of a single input system is shown in Figure 2.3. The closed-loop poles of a position control system using LQ type feedback are plotted as a function of ρ . The system has two open-loop poles at -4.6 and 0.0. As ρ decreases, the closed-loop poles go to infinity along two straight lines that make an angle of $\pi/4$ with the negative real axis. As ρ approaches infinity, both closed-loop poles first meet on the negative real axis and then approach the open-loop poles at (-4.6, 0).

Figure 2.4 shows the asymptotic loci of the closed-loop poles for a multiple input system where $n=4$ and $m=2$. There are four open-loop poles at $(-0.006123, \pm j0.09353)$ and $(-1.250, \pm j1.394)$ and both the state and control weighting matrices are taken to be of a diagonal form. As ρ approaches to zero, one closed-loop pole stay finite at -1.002. The remaining three go to infinity, two of which assume a second order Butterworth pattern and the last one

approaches on the negative real axis. When ρ approaches infinity, all closed-loop poles approach the open-loop poles.

Many researchers have explored this asymptotic properties. Stein [Refs. 9,10] developed a procedure to select the control weighting matrices for a desired closed-loop asymptotic eigenstructure. It will be shown in the subsequent Chapters that the asymptotic properties provide useful guideline for the design procedures to be described.

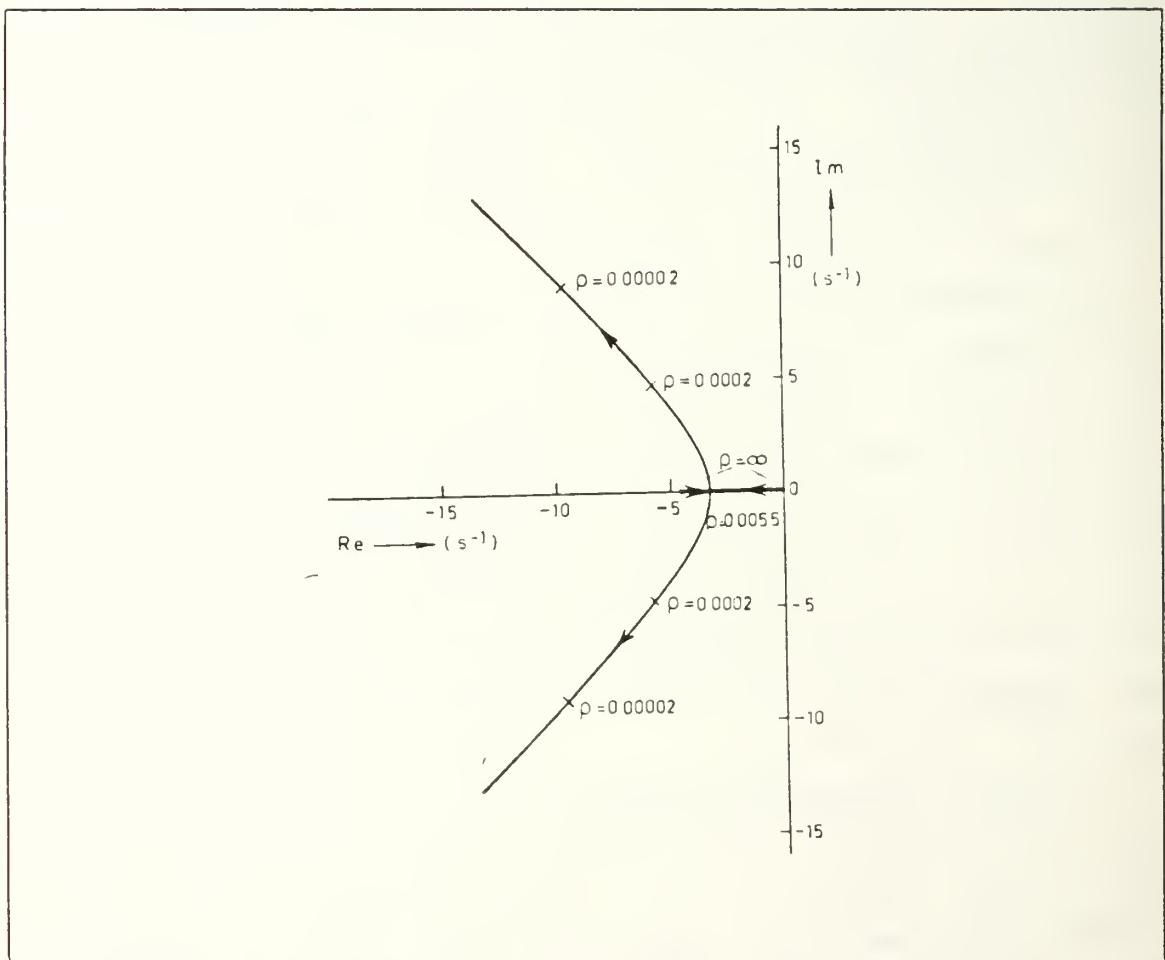
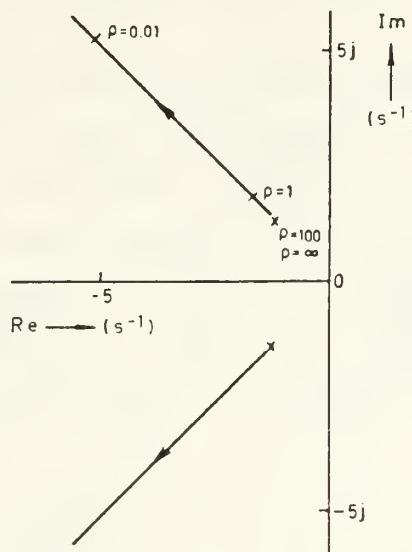
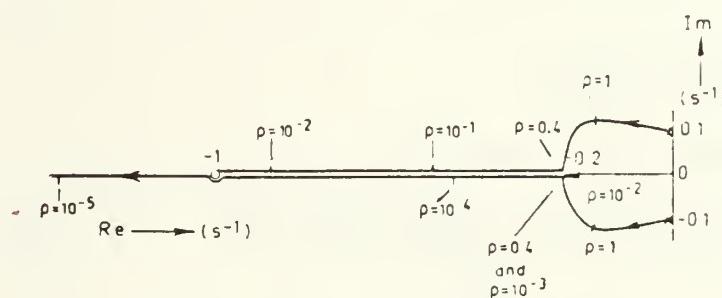


Figure 2.3 Single Input Asymptotic Root Loci.



Faraway Poles



Nearby Poles

Figure 2.4 Multiple Input Asymptotic Root Loci.

III. ROBUSTNESS THEORY AND LINEAR QUADRATIC CONTROL

In the last Chapter, the effect of weighting matrices on the closed-loop pole of the LQ system was discussed. This Chapter addresses yet another important feedback property that control system designers are concerned with: Robustness. Robustness theory was developed when it was realized that the classical single loop Nyquist test was not adequate to guarantee stability when the multivariable open loop plant deviates from its model due to a variety of reasons [Ref. 15]. In the following section , the concept of robust design for both SISO and MIMO general feedback system is reviewed. The singular value analysis is discussed in term of multiplicative type of disturbances. Finally, robustness for linear quadratic state feedback are presented in terms of the effect of weighting matrices on the singular value curve.

A. ROBUSTNESS CONCEPTS

The concept of robustness for SISO system can best be described in terms of the definition of phase and gain margin. A system characterized by good gain and phase margin implies that changes in the plant model parameters and changes in the loop gain and/or phase may be accommodated without loss of stability. The gain and phase margins of a SISO system are defined with reference to the perturbed system in Figure 3.1

Assuming that the unperturbed system ($1(jw)=1$) is stable, the positive (or negative) phase margin is the value of Φ greater (or less) than zero at which the perturbed system with $1(jw) = k \cdot \exp(j\Phi)$ becomes unstable. The upward

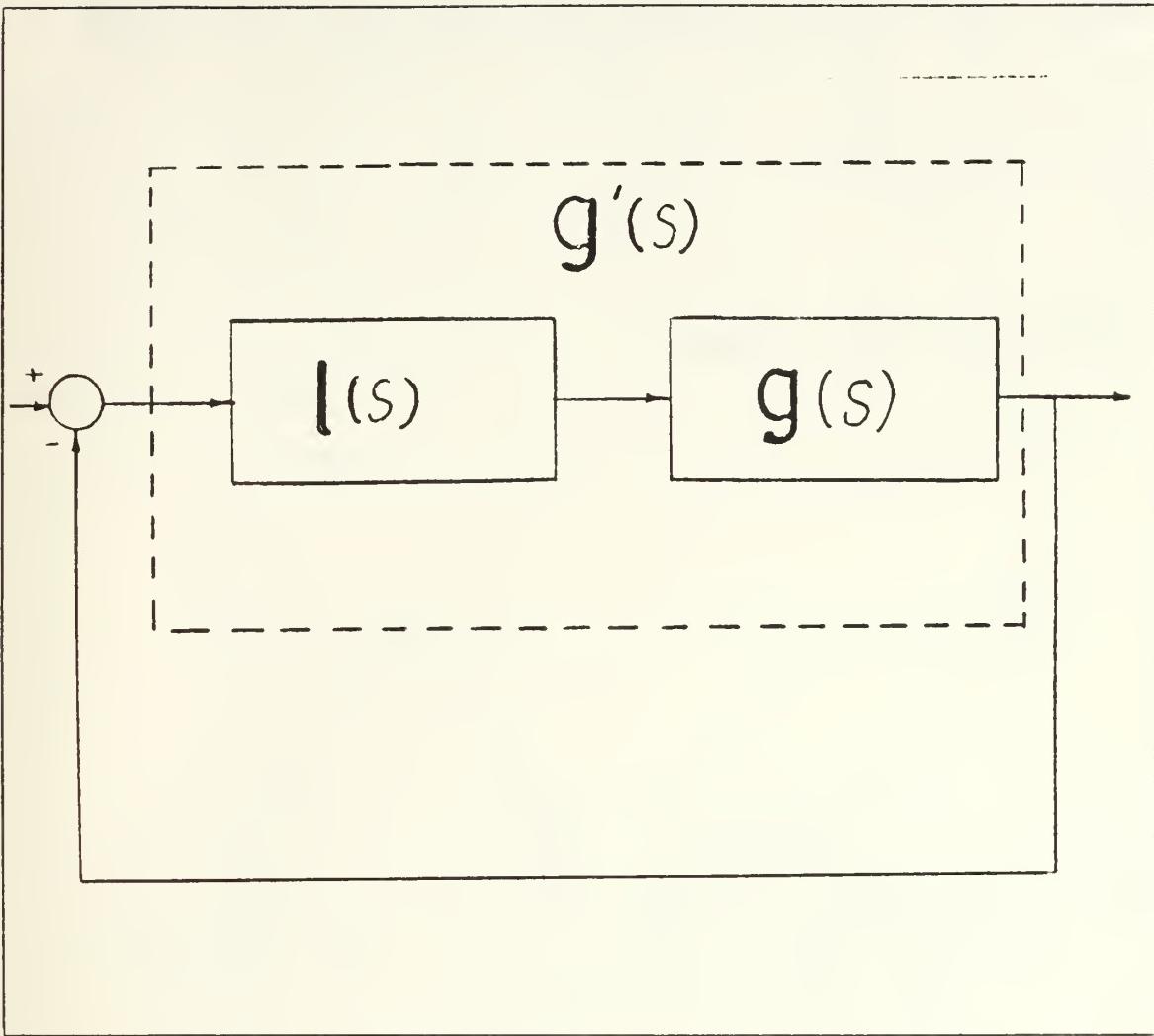


Figure 3.1 The Perturbed SISO System.

(or downward) gain margin is the smallest (or greatest) value of $1(j\omega)=k=\text{constant}$, for which the system become unstable. With reference to the classical Nyquist plot in Figure 3.2 , gain and phase margin are defined as,

$$GM \uparrow (\text{upward}) = 1/k_1 \quad GM \downarrow (\text{downward}) = 1/k_2$$

$$PM^+ = \alpha_1 \quad PM^- = \alpha_2$$

A set of minimum guaranteed GM and PM may be obtained by defining $\alpha_0 = \min[1 + g(j\omega)]$ as shown in Figure 3.3 where,

$$GM = 1/(1 \pm \alpha_0)$$

and

$$PM = \pm \cos^{-1}(1 - \alpha_0^2/2)$$

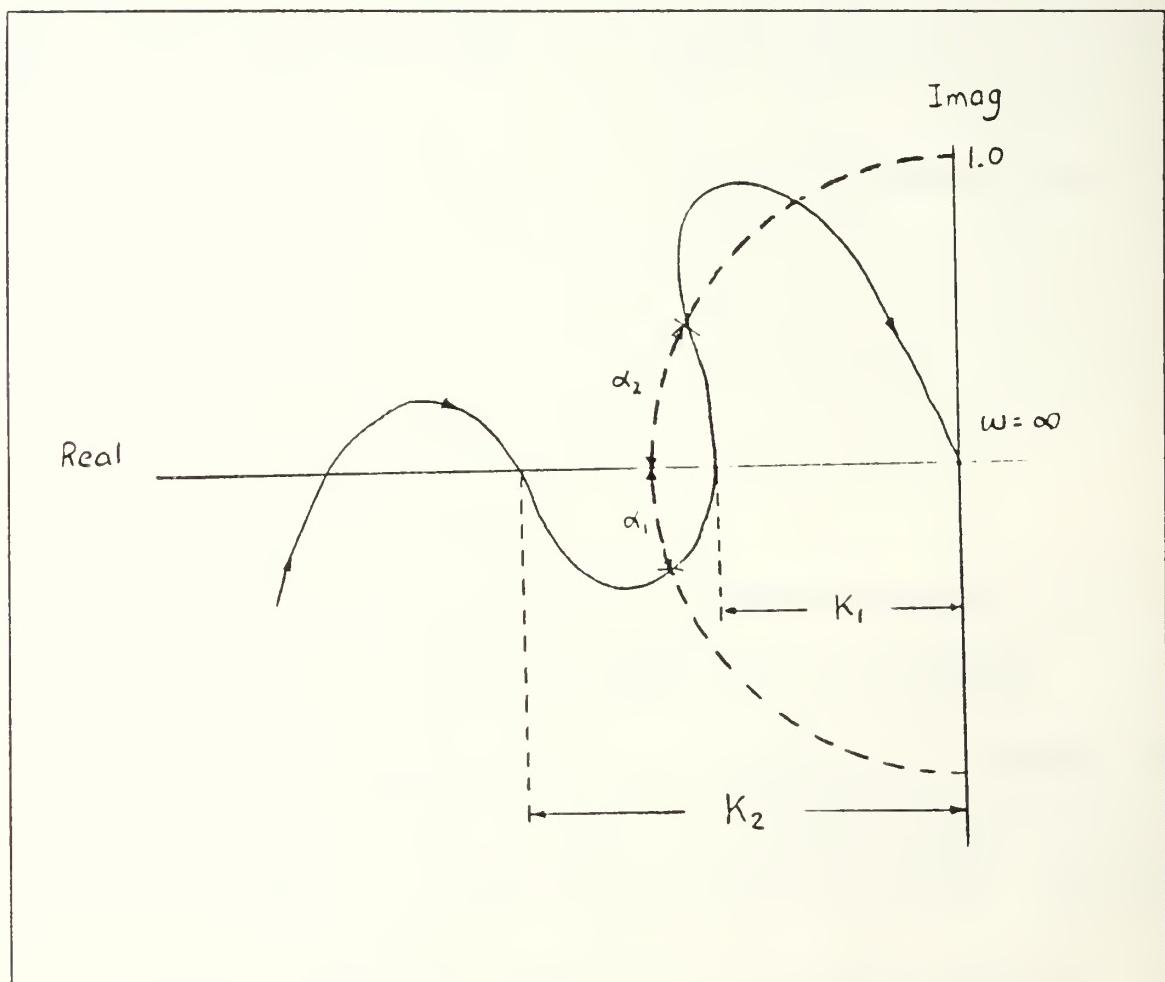


Figure 3.2 The Nyquist Plot - Gain and Phase Margin.

Note that in the above definition of GM and PM, either Φ_i 'or' K_i is allowed to change in the loop. The allowed changes are therefore very restrictive. A more useful definition of the gain and phase margin for MIMO system that accounts for simultaneous changes in both Φ_i and/or K_i (the so-called universal gain and phase margin) has been derived in [Ref. 19]. From Figure 3.3 it can be seen that α_0 may provide a rather conservative estimate of the actual gain and phase margin.

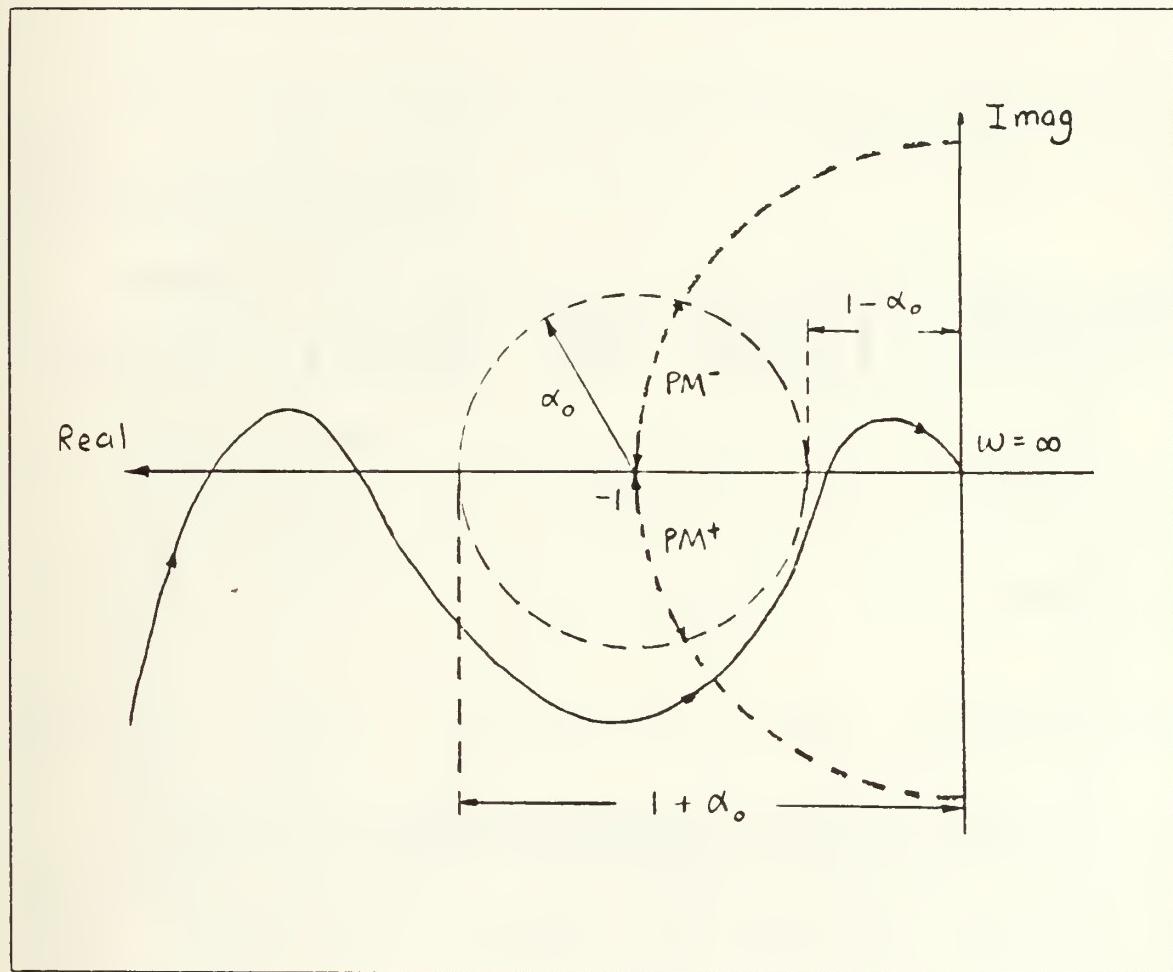


Figure 3.3 The Minimum Guaranteed GM and PM.

In MIMO system, gain and phase margin characterize the ability of the system to tolerate gain and/or phase changes within all loops simultaneously. Figure 3.4 shows a perturbed MIMO system with the assumption that $L(jw) = \text{Dia}(l_1(jw), l_2(jw), \dots)$. As in the SISO case, the system will remain stable as long as $l_i(jw)$ satisfy $GM_i \downarrow < k_i < GM_i \uparrow$ (assuming that Φ_i 's are zero). For the case when the magnitude of $l_i(jw)$ are constant, the system will remain stable for all Φ_i 's that satisfy $PM^- < \Phi_i < PM^+$.

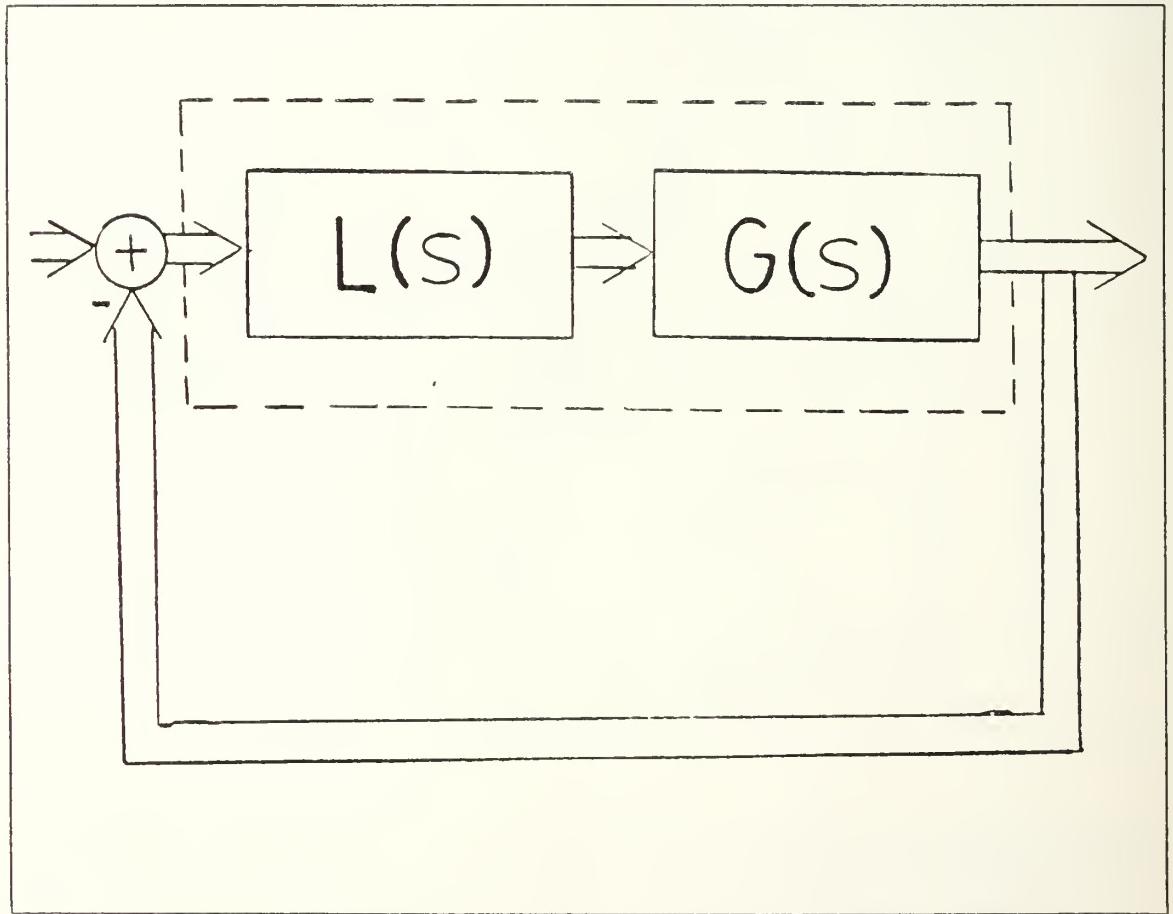


Figure 3.4 The Perturbed MIMO System.

The set of minimum guaranteed multivariable gain and phase margins are determined by the minimum singular value ($\underline{\sigma}$) of the return difference matrix $[I + G(s)]$, where $G(s)$ is the open loop transfer matrix and consists of the plant and its controller. Note the similarity between the SISO and MIMO cases,

α_o : nearness of $(I + g(s))$ to the origin.

$\underline{\sigma}$: nearness of matrix $[I + G(s)]$ to singularity.

Two important results that are related to multivariable phase and gain margin developed in [Ref. 15] are next presented as a theorem.

THEOREM: The multiplicative perturbed system (Figure 3.4) is stable if either of the following conditions hold:

$$1. \quad \underline{\sigma} [I + G(s)] > \bar{\sigma} [L^{-1}(s) - I] \quad (\text{eqn 3.1})$$

$$2. \quad \underline{\sigma} [I + G^{-1}(s)] > \bar{\sigma} [L(s) - I] \quad (\text{eqn 3.2})$$

where $\bar{\sigma}$, $\underline{\sigma}$ denote the maximum and minimum singular values of $[I+G(s)]$ respectively.

It will be shown that condition 1 can be related to the optimality condition of LQ system and hence provides a useful relationship between the weighting matrices and robustness.

B. ROBUSTNESS IN LQ SYSTEM

Robustness properties pertaining to LQ system are closely related to the frequency domain optimality condition. For the SISO case, Kalman [Ref. 1] showed that the return difference transfer function satisfies the inequality

$$(1 + g(jw)) \geq 1 \quad (\alpha_0 = 1) \quad (\text{eqn 3.3})$$

Inspection of the Nyquist diagram in Figure 3.5 clearly indicates that a SISO LQ state feedback has a guaranteed infinite upward gain margin, 0.5 downward gain margin and a minimum phase margin of ± 60 deg.

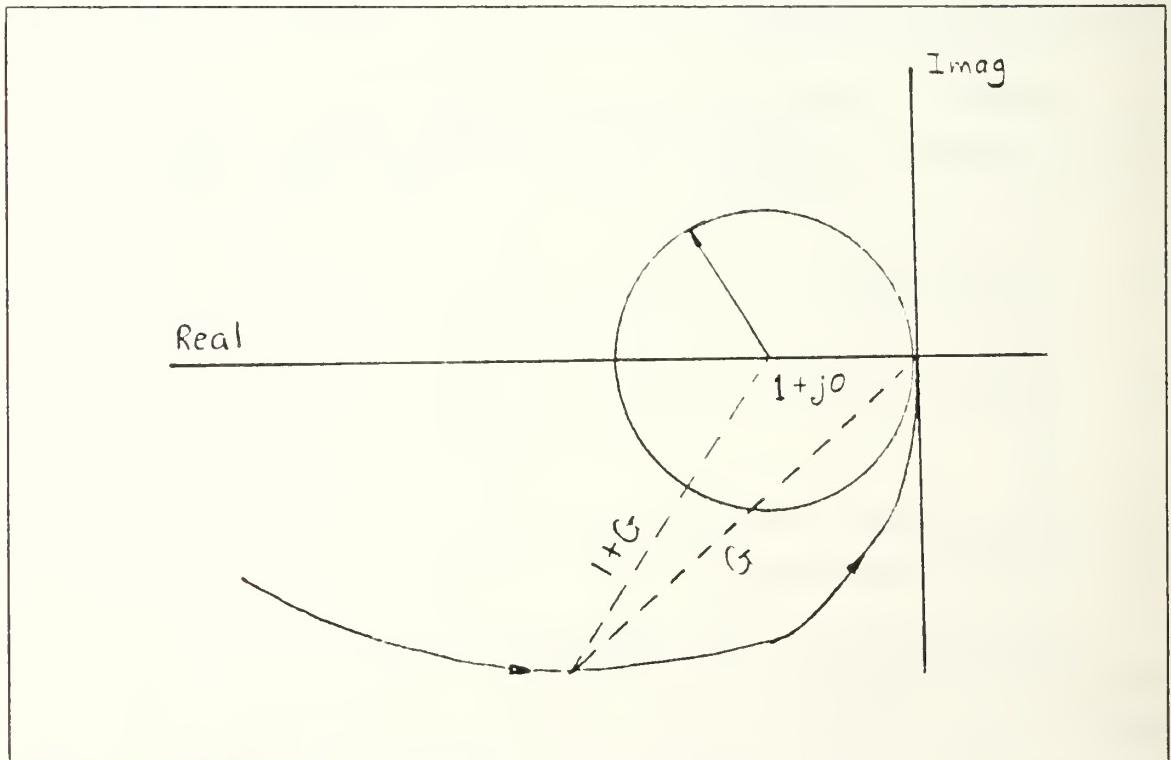


Figure 3.5 SISO LQ Nyquist.

In a similar manner, multivariable stability margin can be derived from the multivariable version of condition 1 in equation 3.1 ;

$$[I + G(s)]^H R [I + G(s)] > R \quad (\text{eqn 3.4})$$

where

$$G(s) = R^{-1} B^T P [sI - A]^{-1} B$$

and $P \geq 0$ satisfies the steady state Riccati equation.

It can be shown that condition 1 in equation 3.1 above can be written as

$$\underline{\sigma} [I + R^{1/2} G(jw) R^{-1/2}] \geq 1 \quad (\text{eqn 3.5})$$

which can be reduced to

$$\underline{\sigma} [I + G(jw)] > 1 \quad (\text{eqn 3.6})$$

where R is diagonal (i.e. $R = \rho I$ for some positive scalar ρ). Like the SISO case, the multivariable LQ regulator with loop transfer matrix $G(s)$ that satisfies equation 3.5 and 3.6 has, in all the feedback loops, a guaranteed minimum gain and phase margin given by,

$$GM = 1/2, \text{ and } PM = \pm 60 \text{ deg}$$

The case when R is not diagonal is interesting as condition 3.6 no longer apply and additional trade-off can be obtained by including off diagonal terms in the control input weighting matrix. This is especially useful when the nature and structure of the disturbances are known. Design involving the off diagonal R matrix will be shown in Appendix A.

IV. DESIGN PROCEDURE

A. GENERAL

The background material described in Chapter 2 and 3 are the basis for developing a computer aided design package and the corresponding design procedure. The design philosophy presented in this Chapter is most useful when a designer is able to characterize the desired system in terms of closed-loop eigenvalues and time response. Some initial physical insight of the weighting matrices ,their asymptotic properties and the nature of the perturbation will be useful in the design process. This Chapter begins with a discussion of the various approaches to the LQ pole placement problem. The selected approach and the corresponding computer aided design package is presented. The design philosophy and design procedure for both the reduced order and full order model are then given.

B. APPROACHES

All the pole placement algorithms for LQ system that have been developed so far require an expression that relate the characteristic equation of the optimal system with the elements of the weighting matrices. Two such formulations for the stabilizable and detectable time-invariant linear system (equation 2.1) and the quadratic criteria (equation 2.3) are given by

$$\phi_c(s)\phi_c(-s) = \phi(s)\phi(-s)\det[I + R^{-1} H^T(-s) Q H(s)] \quad (\text{eqn 4.1})$$

and

$$\phi_c(s)\phi_c(-s) = \phi(s)\phi(-s)\det[I + 1/\rho R^{-\frac{1}{2}} H^T(-s)QH(s)R^{-\frac{1}{2}}] \quad (\text{eqn 4.2})$$

where $\phi_c(s) = \det[sI - A + BF]$ and $\phi(s) = \det[sI - A]$ are the closed-loop and open-loop characteristic polynomials respectively. $H(s) = C(sI - A)^{-1}B$ is the open loop transfer matrix of the system.

Both formulations have been used in root-locus type of procedure to investigate how the closed-loop poles move as weighting matrices changes [Refs. 5,6]. For MIMO case, there has been little success due to problems involving polynomial matrices. In this work, a different approach is adopted. Equation 4.1 or 4.2 is formulated as a numerical optimization problem in which the objective function is made equal to the determinant part of equation 4.1 or 4.2

$$\text{Obj} = \det[I + R^{-\frac{1}{2}} H^T(s) Q H(s)] \quad (\text{eqn 4.3})$$

or

$$\text{Obj} = \det[I + 1/\rho R^{-\frac{1}{2}} H^T(-s) Q H(s)R^{-\frac{1}{2}}] \quad (\text{eqn 4.4})$$

For a given desired closed loop pole $s = s_d$, equation 4.3 or 4.4 becomes

$$\phi_c(s_d) \phi_c(-s_d) = 0 \quad (\text{eqn 4.5})$$

Providing that $\phi(s_d)\phi(-s_d)$ is not equal to zero, the objective function must equal to zero if the particular Q , R set is to correspond to the desired closed-loop poles. Convergence to zero for a given set of Q and R is therefore automatically guaranteed.

The pole placement problem can therefore be solved as an unconstrained multivariable optimization in which the elements of Q and R are varied to make the objective

function in equation 4.3 and 4.4 approach zero. This was done during the early phase of the work. It was later discovered that more insight to the problem can be obtained by first transforming the problem to an appropriate coordinate system and then to perform pole placement one at a time. This is of advantage as a system designer is often satisfied with several open-loop poles in a large system. Reassigning poles in the the reduced-order model will reduce computer time and memory requirement.

The optimization routine selected for this work is the so-called SUMT method (Sequential Unconstrained Minimization Techniques) obtained from the ADS package in [Refs. 20,21]. In this method, the objective function (eg. equation 4.3 or 4.4) and any constraint equation are formulated into an augmented objective function in which the problem is solved as an unconstraint optimization task.

C. POLE PLACEMENT ALGORITHMS

For ease of implementation and better insight, only the problem of determining the state weighting matrix Q (given R) that gives a set of closed-loop eigenvalues is considered. It must be emphasised that the algorithm can also be formulated to determine R (for a given Q) or to vary Q and R at the same time. In most cases, the present formulation is adequate as designers usually have some knowledge about the control weighting matrix. System matrix that has real and distinct eigenvalues is presented first, follows by cases where A has complex eigenvalues and repeated eigenvalues.

1. System with Real and Distinct Eigenvalues

The original system given by equation 2.1 and 2.2 is first transformed into a diagonal form using the transformation given by;

$$x(t) = Mz(t) \quad (\text{eqn 4.6})$$

where M is an eigenvector matrix corresponding to the system matrix A . and $z(t)$ is the new state vector. The transformed system in the new coordinate is given by,

$$\dot{z}(t) = \Lambda z(t) + M^{-1}Bu(t) \quad (\text{eqn 4.7})$$

where Λ is a diagonal matrix $\text{dia}[\lambda_1, \lambda_2, \lambda_3, \dots]$

The performance index (equation 2.3), when expressed in terms of the new state vector $z(t)$ becomes,

$$\begin{aligned} J &= \int_0^\infty (x^T Q x + u^T R u) dt \\ &= \int_0^\infty (z^T M^T Q M z + u^T R u) dt \\ &= \int_0^\infty (z^T \hat{Q} z + u^T R u) dt \end{aligned} \quad (\text{eqn 4.8})$$

where $\hat{Q} = M^T Q M$.

It can be shown that to move an open-loop pole to its new location given by $s_i = s_d$, only \hat{Q}_i is required and other \hat{Q}_j 's have no effect on the pole assignment. As an example, to move the open-loop pole at $s = \lambda_2$ for $\Lambda = \text{dia}[\lambda_1, \lambda_2, \lambda_3, \dots]$ to its new location $s = \bar{\lambda}_2$, only $\hat{Q} = \text{dia}[0, \hat{q}_1, 0, \dots]$ is required.

\hat{Q} can then be selected according to equation 4.3 or 4.4 using the optimization routine. As currently implemented in the program, there is no constraint equation formulation. Once the value of \hat{Q} that satisfies the desired pole location is obtained, the system is transformed back to the original coordinate system via,

$$Q = M^{-T} \hat{Q} M^{-1} \quad (\text{eqn 4.9})$$

With Q known, and R given, the optimal feedback gain F can be obtained by solving the steady state Riccati equation as given in equation 2.5 and 2.6 . Since the pole placement is done in the decoupled coordinate system, only the eigenvalues that correspond to Q_i is reassigned; all other eigenvalues remain unchange. It can be also shown that the eigenvector corresponding to an eigenvalue is also unchange, this property will be shown to be useful in the reduced order formulation of the linear quadratic problem.

If desired, the problem here can also be formulated to move more than one eigenvalues in one run. This can be done by modifying the objective function to include more terms as follows;

$$\text{Obj} = \sum_{i=1}^n \det[I + R^{-1} H^T(-s) Q H(s) R^{-1}] \quad (\text{eqn 4.10})$$

where n is the number of poles to be reassigned.

The augmented matrix $A_{\text{aug}} = [A + BF]$ is then computed. If desired, the procedure may be repeated to move other open loop eigenvalue to its specified position using the new A_{aug} as the starting plant matrix. This will in turn result in another set of Q and F . The effective Q_e and F_e after n reassignments are given by

$$Q_e = Q_1 + Q_2 + \dots Q_n \quad (\text{eqn 4.11})$$

and

$$F_e = F_1 + F_2 + \dots F_n \quad (\text{eqn 4.12})$$

The above pole assignment procedure can also be applied to an optimal system where an initial starting Q and R are given. A good example is when the control system designer has some knowledge of the weighting matrix but would also like to meet a specific time response requirement.

2. System with Complex Eigenvalues

If the same similarity transformation mentioned in the last section is used, the transformation matrix will be complex. To be able to work with real matrix, an auxiliary transformation of the form given by equation 4.13 is used;

$$x(t) = Tz(t) \quad (\text{eqn 4.13})$$

$T = ML$ and M is the eigenvector matrix (equation 4.6) The matrix L is given by,

$$L = \begin{bmatrix} 0.5 & -0.5j & 0.0 & \dots & 0.0 \\ 0.5 & 0.5j & 0.0 & \dots & . \\ 0.0 & 0.0 & 1.0 & \dots & . \\ . & \dots & \dots & 1.0 & . \\ 0.0 & \dots & \dots & \dots & 1.0 \end{bmatrix} \quad (\text{eqn 4.14})$$

The transformed system is then given by,

$$\dot{z}(t) = A z(t) + T^{-1}Bu(t) \quad (\text{eqn 4.15})$$

with performance index ,

$$J = \int (z^T T^T Q T z + u^T R u) dt \quad (\text{eqn 4.16})$$

where \hat{Q} is now given by $\hat{Q} = T^T Q T$.

It can be shown that to move a pair of complex eigenvalues given by $s = a + bj$, a weighting matrix of the form

$$\hat{Q} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & \hat{q}_i & 0 & & \cdot \\ 0 & 0 & \hat{q}_{i+1} & & \cdot \\ 0 & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \dots & 0 \end{bmatrix}$$

(eqn 4.17)

with $\hat{q}_i = \hat{q}_{i+1}$ is required.

In a similar manner, \hat{Q} can be obtained by using the optimization routine, with the condition $\hat{q}_i = \hat{q}_{i+1}$ formulated as a constrained equation. Inverse transformation and determination of Q_e and F_e are identical to the distinct eigenvalue case with M in equation 4.9 replaced by T .

3. System With Repeated Eigenvalues

In this case, the system matrix cannot be diagonalized but the general procedure given above still apply. The system is first transformed into the Jordan canonical form ;

$$J = U^{-1} A U \quad (\text{eqn 4.18})$$

where U is a transformation matrix which is not the eigenvector matrix M . An example of the Jordan form is given below,

$$J = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}$$

(eqn 4.19)

The only difference with the two procedures mentioned above is that the pole reassignment has to begin at the bottom of each Jordan block. For example, in the system given above (equation 4.19), the first re-assignment will result in a new system given by equation 4.20

$$J = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & s_d \end{bmatrix}$$

(eqn 4.20)

Determination of Q and F using the optimization and the inverse transformation routines are identical to the distinct eigenvalues case with M in equation 4.9 replaced by U.

D. COMPUTER PROGRAM DESCRIPTION

The computer aided design package developed in this thesis is illustrated in Figure 4.1. The program is developed using the top-down approach with special purpose subroutines called by the main control program. The driver program supports three independent modes with the data entry portion common to all modes:

1. Data Entry : The system matrices A, B, C and/or F, Q, R etc are entered through a data file. Design variables such as desired poles locations, elements of matrices to be varied etc, are also specified through the input data file.
2. Pole Placement Mode : In this mode, an arbitrary set of closed-loop eigenvalues is assigned by selecting the appropriate state weighting matrix. As shown in Figure 4.1, the transformation matrices for various

- cases are computed first. The pole placement is achieved using the numerical optimization routine described in the last section; Q is obtained and then inverse-transformed to the original co-ordinate system. If desired, the results can be used as an input to the Linear Quadratic Control Program.
3. Linear Quadratic Control Mode : This part of the program is adopted from the OPTSYS program. Given a set of weighting matrices Q and R and the system matrices A, B and C, it computes the steady state feedback gain F, closed-loop eigenvalues, etc.
 4. Singular Value Analysis Mode : This portion of the program allows the designer to analyze various designs obtained from the two modes mentioned above in term of singular value vs frequency plot. The main part of this program is adopted from [Ref. 22].

The three modes of operation mentioned above may be used in any order to implement specific design objectives. A typical design process will involve runs alternating between the three modes until a compromise between primary and secondary design objectives is achieved. Record of a typical design run together with a complete listing of the main program and their non-standard subroutine are given in Appendix B and C.

E. DESIGN PHILOSOPHY

The Linear Quadratic constant state feedback design philosophy for the linear time-invariant model is illustrated in Figure 4.2. It assumes that the location of the eigenvalues and system time response are the main design objectives. These objectives can be achieved using the pole placement procedure developed in this thesis. Single or multiple reassignment can be made in one run. A number of

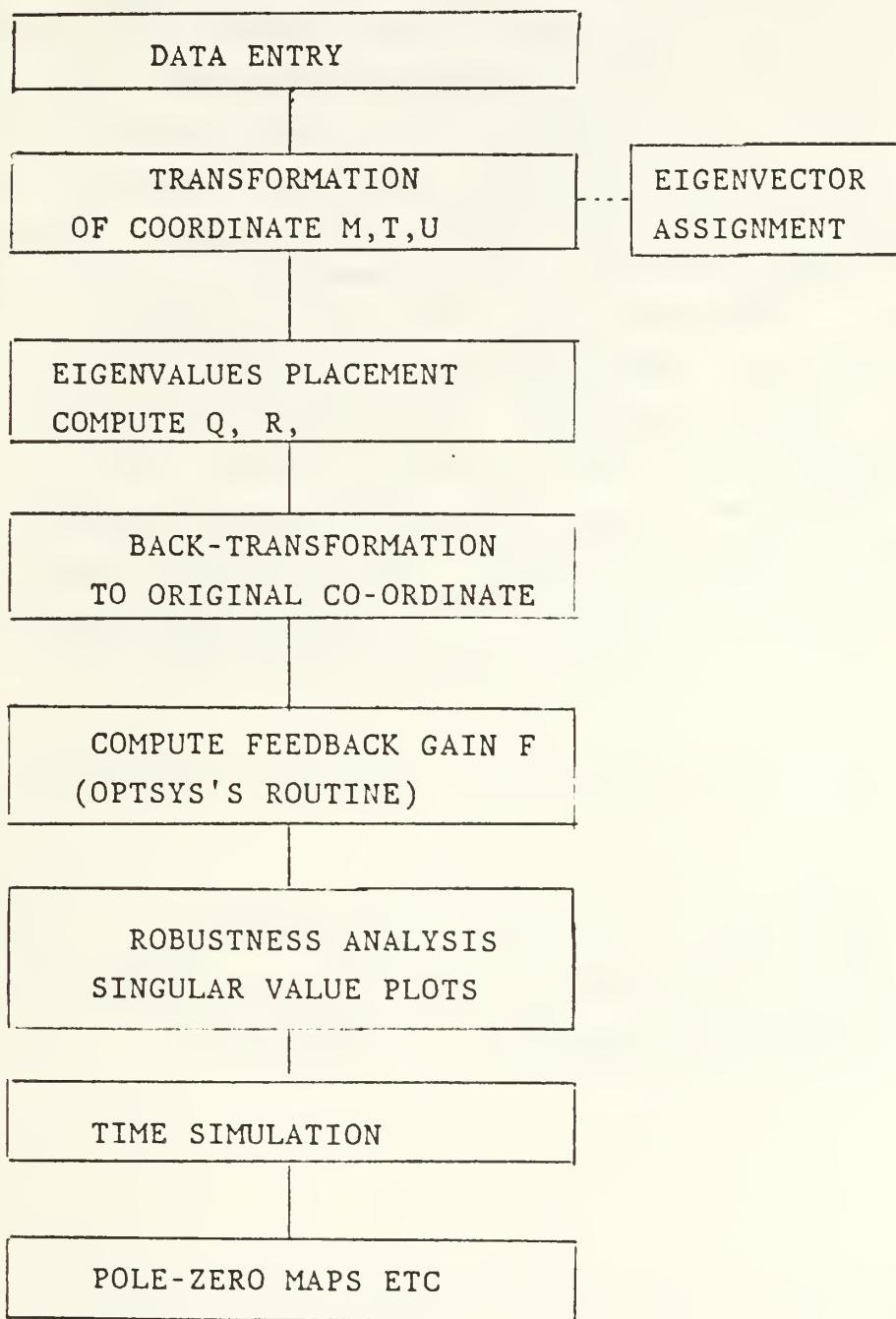


Figure 4.1 Computer Aided Design Package Organization.

designs can then be obtained using different starting control weighting matrices, different state weighting matrices and different assignment sequences. Physical constraints such as control input amplitude, control input energy as well as general system properties such as asymptotic behavior are heavily relied upon during this process. After the major objectives are satisfactorily achieved, secondary design objectives are considered. These include feedback gain reduction (by increasing the control weighting matrix), robustness (in terms of minimization of system sensitivity to modelling errors and/or parameter variations), zeros locations, eigenvectors assignments. The extra degrees of freedom available in the MIMO state feedback system can often provide a means to improve these secondary objectives while only slightly modifying the closed-loop pole assignment and thus the time response.

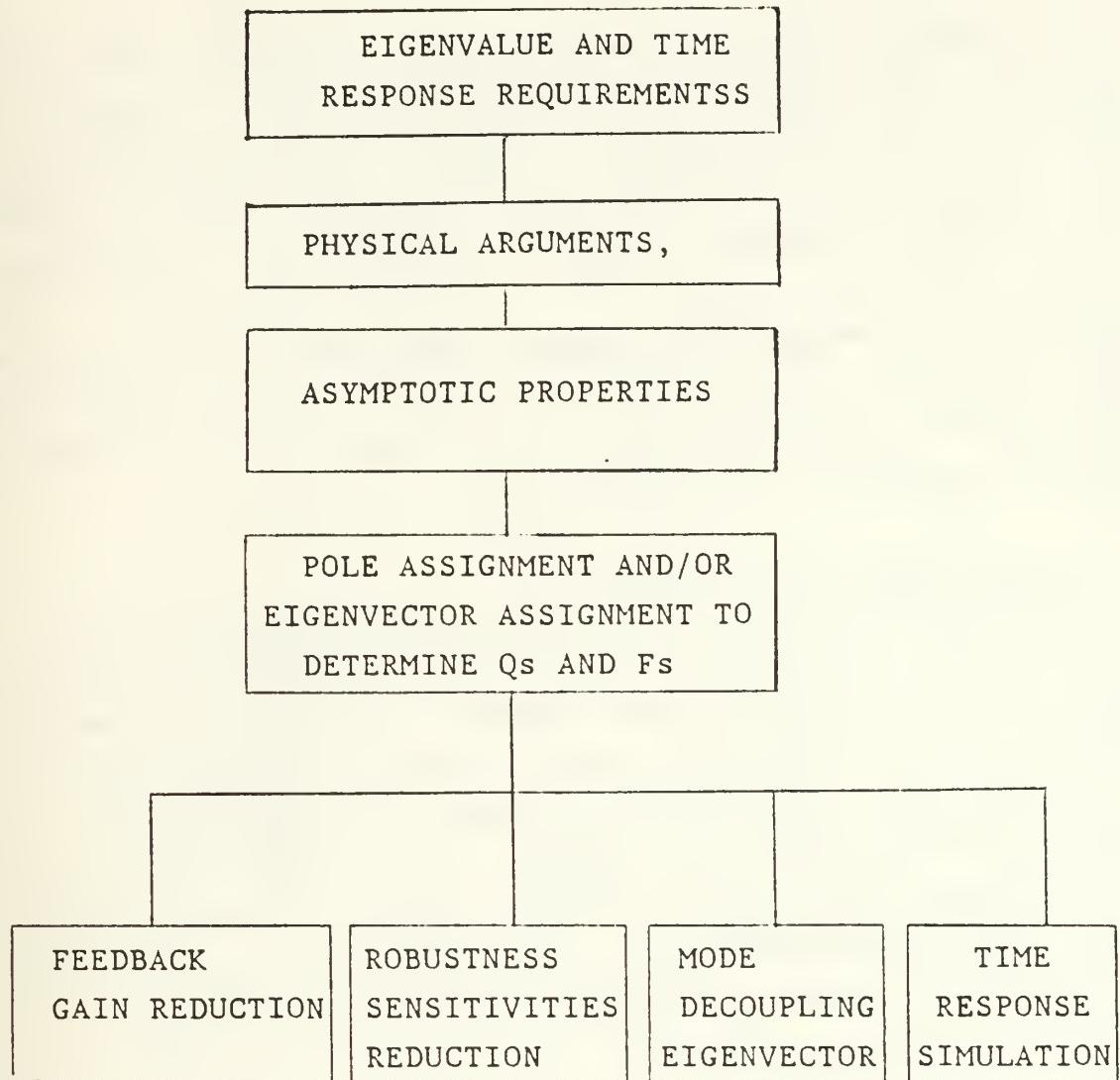


Figure 4.2 LQ Eigenvalue Assignment Philosophy.

V. DESIGN EXAMPLES

The design procedure described in Chapter 4 is illustrated in this Chapter by two design problems. A typical LQ structure and its properties, together with the pole placement procedure is first presented with a 2x2 model. A practical design problem is then presented for the highly coupled lateral channels of a CH-47 Helicopter. The resulting LQ designs are compared with other multivariable state feedback designs [Refs. 22,23]. It is shown that the procedure developed here is a viable tool for robust constant feedback controller design.

A. INTRODUCTORY 2X2 PROBLEM

This problem formulated in reference 23 serves to demonstrate how a highly cross-coupled multivariable control problem can be formulated and solved as a linear quadratic design problem, using the pole placement procedure. The problem provides excellent insight into the structure of the multivariable LQ system and its built-in robustness to cross-coupled perturbation.

Figure 5.1 shows a diagram of this basic 2X2 system in which the plant is given by

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (\text{eqn 5.1})$$

$$y(t) = Cx(t) \quad (\text{eqn 5.2})$$

where

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 50 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

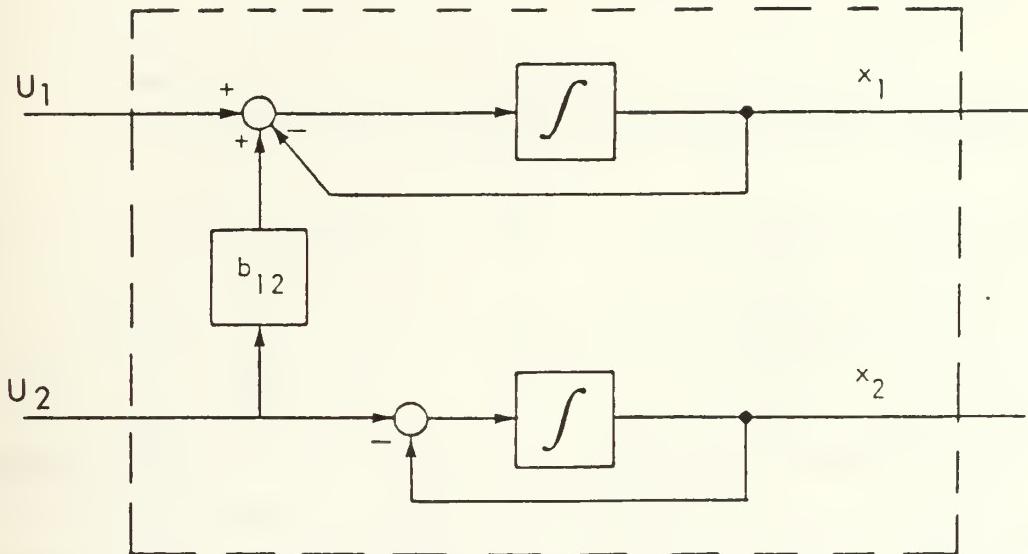


Figure 5.1 A Simple 2X2 MIMO Model.

This system has open-loop eigenvalue at -1, -1 and is therefore stable. The b_{12} term in the control matrix B is purposely made large to produce the cross-coupled effect from channel two to channel one (see Figure 5.1). The design requirement is to select a set of feedback gain F such that the closed-loop eigenvalues are at -2, -2. Assuming that this is the only requirement, it is not difficult to see that a unity feedback law of the form (Figure 5.2),

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_{\zeta 2} \end{bmatrix} \quad (\text{eqn 5.3})$$

will produce the desired closed-loop system given by,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & 50 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 50 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{\zeta 1} \\ u_{\zeta 2} \end{bmatrix} \quad (\text{eqn 5.4})$$

The above design seems to be acceptable as far as eigenvalues or time response is concerned. It is now shown that when robustness of the system is considered, the unity feedback gain controller performs rather poorly. On the other hand, design using LQ Pole Placement type of formulation will result in robust controllers. To demonstrate the lack of robustness of the unity feedback design, the feedback gain matrix F as given in equation 5.4 is perturbed slightly (by +5%) and the eigenvalues of the resulting closed-loop

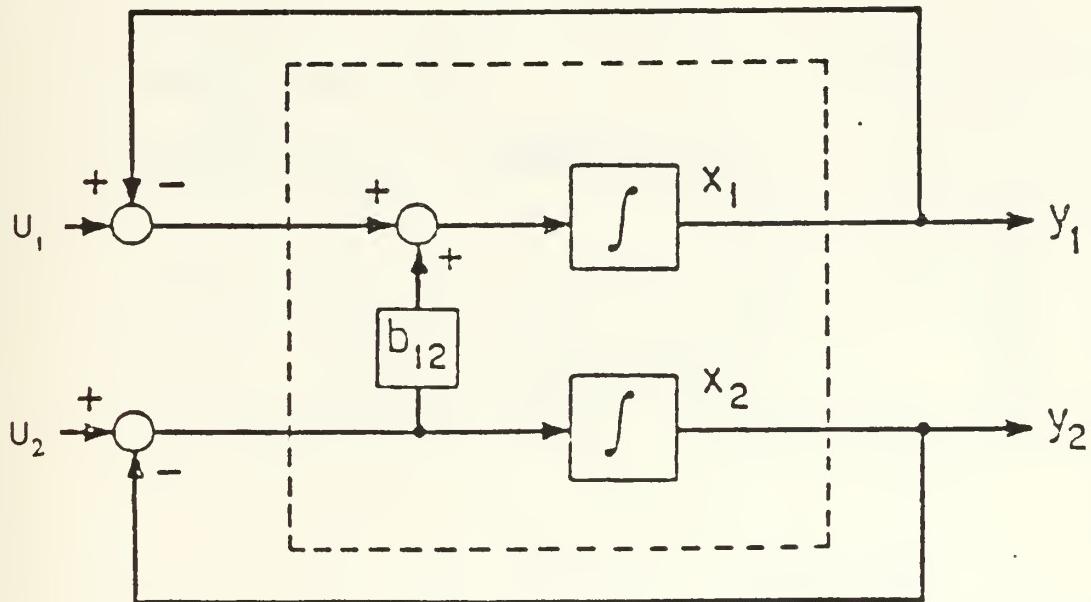


Figure 5.2 Unity Feedback for the 2X2 Model.

system matrix are calculated. The absolute and percentage errors in the eigenvalue due to the perturbation of the closed-loop system are given in Table I .

It can be seen from the above that the first cut design is very susceptible to model and feedback perturbation. A 5 % changes in the f_{21} term has resulted in a large shift (160%) in the closed-loop eigenvalue. Lack of robustness in the unity feedback design can also be seen in terms of the minimum singular value plot of the return difference matrix in the frequency domain. For the unity feedback gain system shown in Figure 5.2, the return difference matrix is given by,

TABLE I
PERTURBED EIGENVALUE FOR 2X2 MODEL

Perturbation in F (+5%)	Absolute Changes in Eigenvalues	Percent Changes in Eigenvalues
f_{11} and f_{22}	0.05 , 0.05	2.5 , 2.5
f_{12}	0.0 , 0.0	0.0 , 0.0
f_{21}	0.67 , +3.265	33.5 , 163.0
All	1,255, 5.346	37.25 , 167.0

$$I + G(s) = \begin{bmatrix} s+2/(s+1) & 50/s+1 \\ 0 & s+2/(s+1) \end{bmatrix}$$

(eqn 5.5)

where $G(s) = C(sI - A)^{-1}B$ is the loop transfer matrix as indicated in Figure 5.3

The multivariable Nyquist diagram (locus of $\det[I+G(s)]$) for the system is shown in Figure 5.4. If this diagram is interpreted as for a single input system, the $(-1/2, \infty)$ gain margin and (± 160) phase margin would lead one to believe that the design is a good one. This has been shown to be not the case, a 5% perturbation in F would cause the system to become unstable. The above clearly demonstrate the inadequacy of the classical method in evaluating stability margin for MIMO system.

Robustness properties of the unity feedback controller will now be analyzed in terms of singular value as discussed

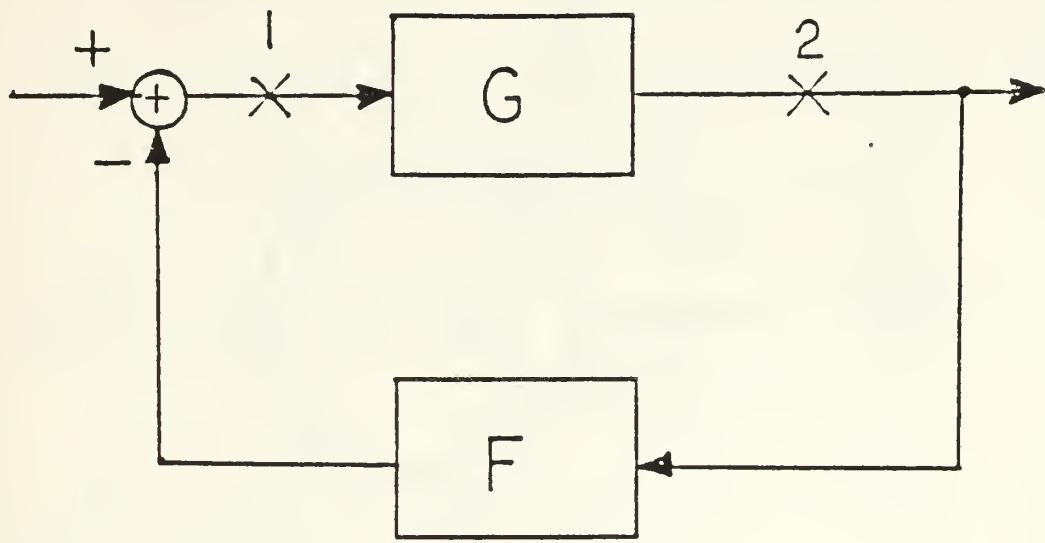


Figure 5.3 Loop Transfer Matrix - $F = I$.

in Chapter 3. The minimum singular value plot of $[I+G(s)]$, is shown in Figure 5.5 as a function of frequency. The lack of robustness is clearly indicated by the relative small singular value at frequency around 2 rad/s. Using the universal phase and gain margin chart developed in [Ref. 19], the minimum singular value at this frequency corresponds to a gain margin of (0.91, 1.0) and a phase margin of (± 4 deg).

It is now shown that formulation using LQ approach and the pole placement procedure developed here will result in robust design that meet the time response requirement. Furthermore, better insight of the design process can be obtained from the procedure to be described here. The first step in the LQ design is to determine the asymptotic

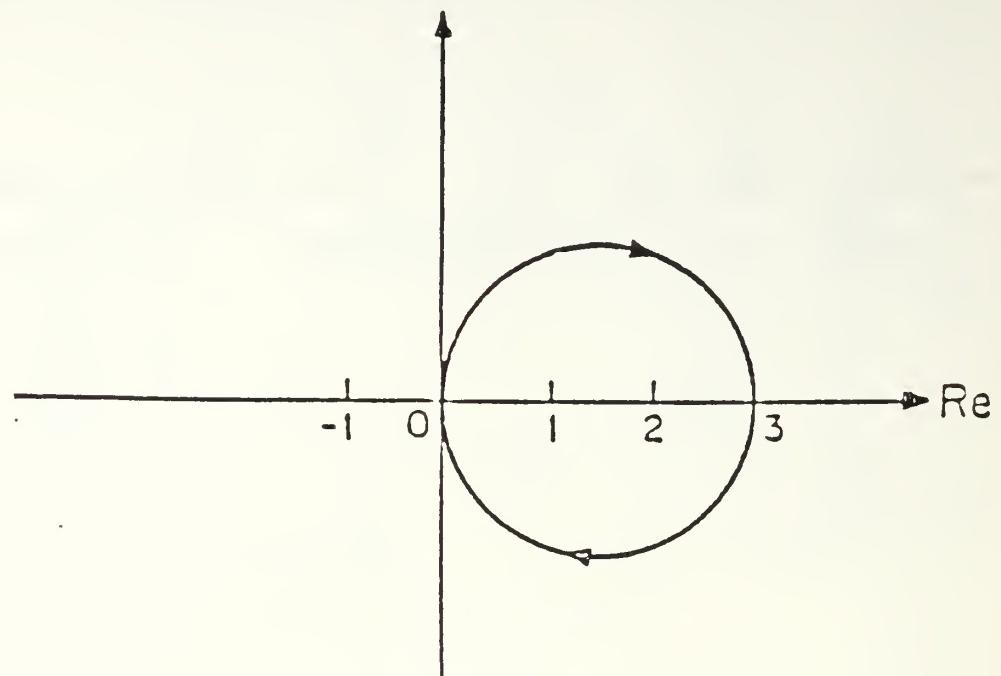


Figure 5.4 Multivariable Nyquist Plot.

properties of the system, i.e. movement of closed-loop poles as $R \rightarrow 0$. Using results from Chapter 2, it can be established that as R increase from 0 to ∞ , both closed-loop poles move from infinity on the real axis to the open-loop poles location. None of the closed-loop poles stay finite as $R \rightarrow 0$ since the dimension of the input control vector is equal to the dimension of the state. Assuming that $R = I$, the pole placement is accomplished in two steps. The first step is to move the open-loop pole at -1 to -2.0. As the system matrix A given is already in Jordan form, no transformation is required. The pole placement program puts the pole at -1.9987 with,

SINGULAR VALUE PLOT

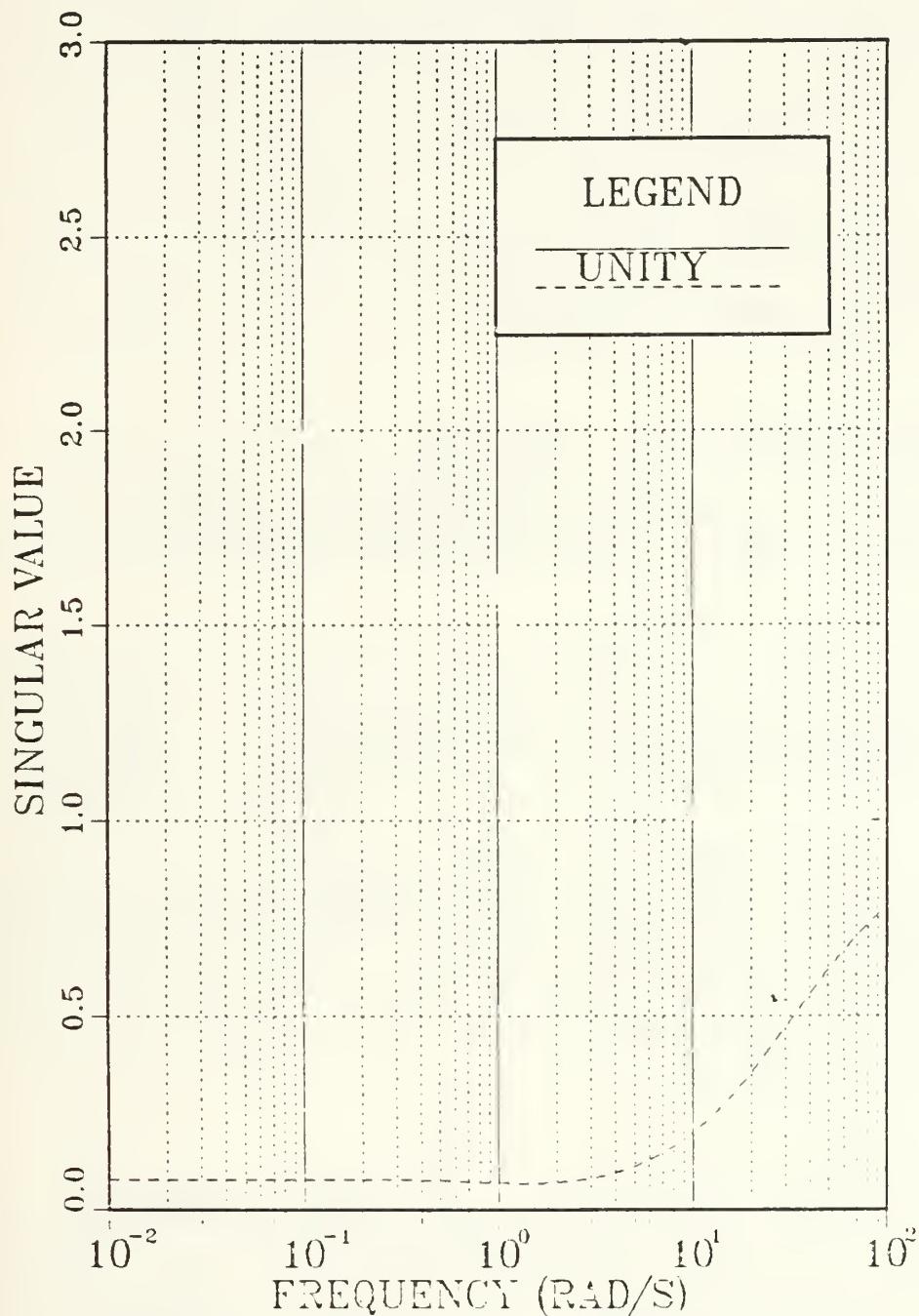


Figure 5.5 Singular Value Plots - Unity Feedback.

$$Q_1 = \begin{bmatrix} 0.00119 & 0 \\ 0 & 0 \end{bmatrix} \quad F_1 = \begin{bmatrix} 0 & 0 \\ 0.01983 & 0 \end{bmatrix}$$

and

$$A_{\text{aug}} = \begin{bmatrix} 1.9915 & 0 \\ 0.01983 & 1.0 \end{bmatrix}$$

In the second step, the other open-loop pole at -1. is moved to -2.0 , using A_{aug} as the new plant matrix. The resulting Q and F and the augmented plant matrix become,

$$Q_2 = \begin{bmatrix} -2.998 & -149.9 \\ -149.9 & 7499.3 \end{bmatrix} \quad F_2 = \begin{bmatrix} 0.9994 & -49.978 \\ -0.00841 & 0.4211 \end{bmatrix}$$

The effective Q_e and F_e required to move both open-loop poles at -1.0 to -2. are $Q_e = Q_1 + Q_2$ and $F_e = F_1 + F_2$ as shown in equation 5.6 below. The pole placement procedure is completed with the final eigenvalues placed at (-1.99255, $\pm j0.05628$).

$$Q_e = \begin{bmatrix} 2.99919 & -149.9 \\ -149.9 & 7499.3 \end{bmatrix} \quad F_e = \begin{bmatrix} 0.9994 & -49.978 \\ 0.0114 & 0.4211 \end{bmatrix}$$

(eqn 5.6)

The singular value plots of the case where $R = I$ together with the unity feedback gain (non-LQ design) are shown in Figure 5.6. The well established fact that LQ design possesses $(1/2, \infty)$ gain margin and $(\pm 60\text{deg})$ phase margin can also be readily observed from the same figure as the minimum singular values of $[I+G]$, $\underline{\sigma}$, is greater than one for all frequency. Changes in closed-loop eigenvalue for a small perturbation in F is again computed as shown in Table II. It can be seen that the LQ design is robust with the largest percentage change in eigenvalues location of only 10%, when compared with the 160% change in the unity feedback design.

TABLE II
PERTURBED EIGENVALUE FOR LQ DESIGN

Perturbation in F (+5%)	Absolute Changes in Eigenvalues	Percent Changes in Eigenvalues
f_{11} and f_{22}	0.2149, 0.142	10.7, 7.1
f_{12}	0.119, 0.119	5.59, 5.59
f_{21}	0.0592, 0.0578	2.90, 2.89
All	0.07, 0.070	3.52, 3.52

The pole-zero plots of various closed-loop transfer functions of the closed-loop transfer matrix for both the unity feedback and LQ design are compared in Figure 5.7 and 5.8. For the unity feedback design, zero at minus three for the input 2 to output 1 channel corresponds to the

SINGULAR VALUE PLOT

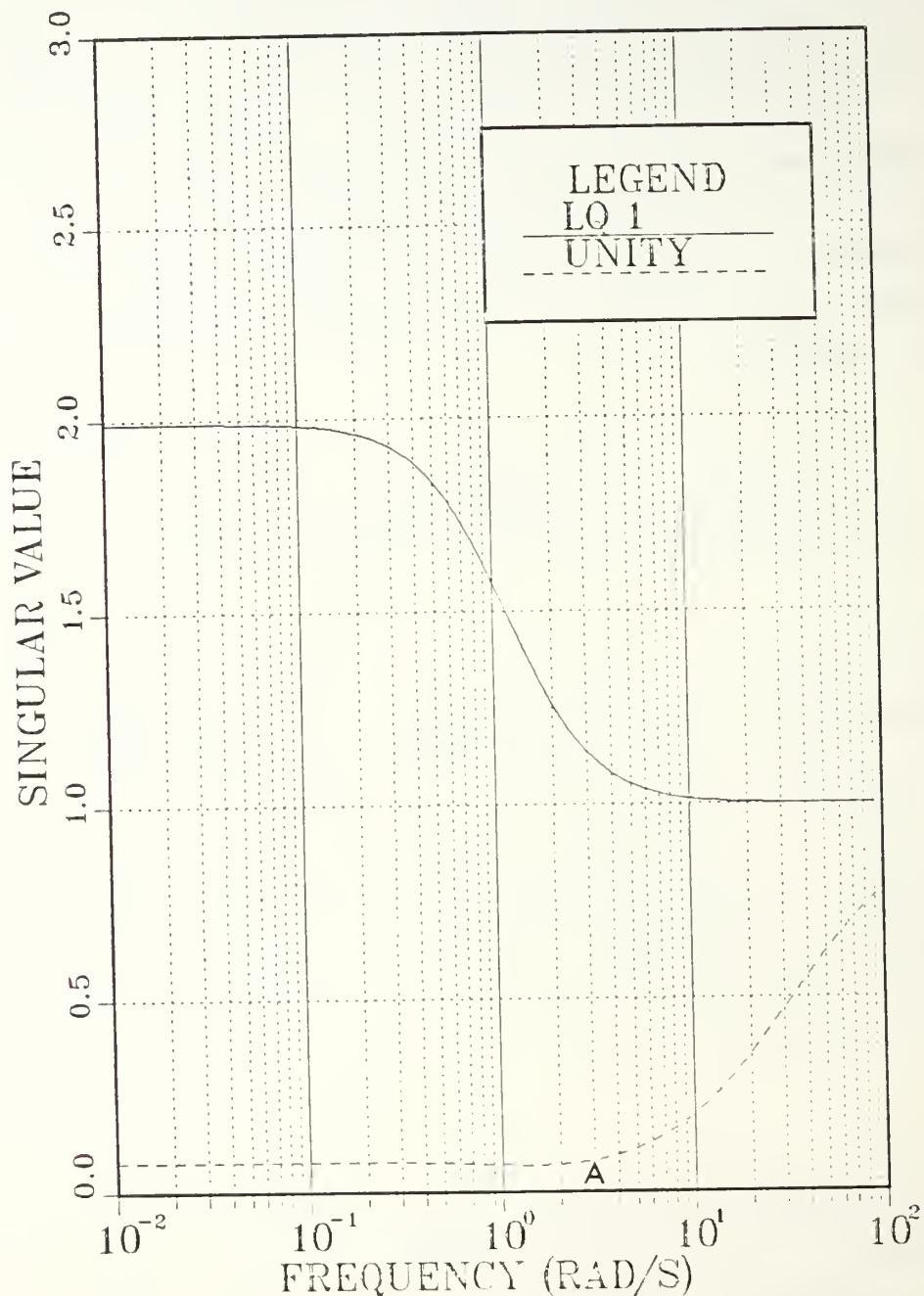
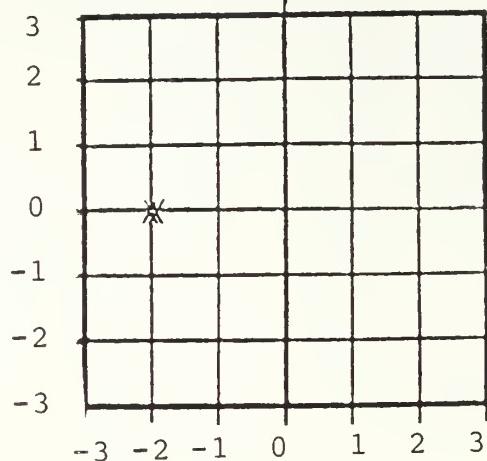


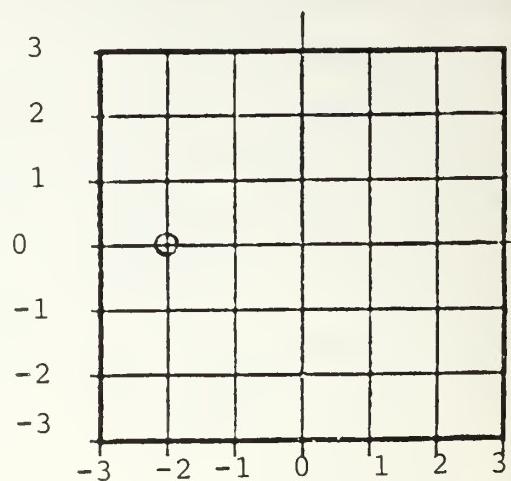
Figure 5.6 Comparison-Singular Value Plots (2x2 model).

minimum singular value frequency (point A in Figure 5.6). It is in fact this zero that causes the the system to be sensitive to perturbation. For the LQ design, the built in robustness cause the zero at minus three to move toward the pole location at -2.

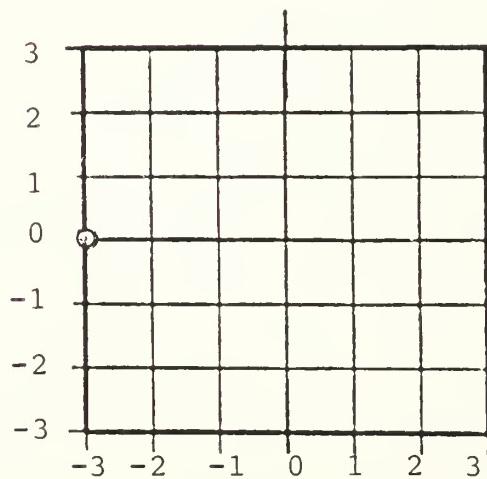
Improvement in the robustness from LQ design is now analyzed in terms of Bode plots. The open-loop gain and phase vs frequency plots for a MIMO system can be obtained from the open-loop transfer function matrix, $G(s)$. For full state feedback, $G(s) = F(sI - A)^{-1}B$. (if the open-loop plots without feedback is considered, $G(s) = C(sI - A)^{-1}B$) The matrix G is a matrix-valued rational function of s , it describes how the system (with or w/o feedback) appears to its environment. It is an external description of the system and is closely related to the zeros of the system. The open-loop Bode plots for the two designs are compared in Figures 5.9 to 5.11 As $b_{21} = 0$, there is no coupling from channel 1 to channel 2. All channels have a -20 dB/decade slope at high frequency which is in agreement with earlier observation that each channels has one finite zero. It is interesting to note that the unity feedback and LQ design result in almost identical gain vs frequency plots for direct channels (i.e. channel 1-1 and 2-2). Any classical single loop type of analysis will not be able to detect any difference between the two designs. On the other hand, cross-coupling effect can be readily seen from the gain vs frequency plot in channel 2-1 (Figure 5.10). The unity feedback design is characterized by the rather large channel 2-1 gain at low frequency. A very small perturbation in channel one's parameter can change the system behavior considerably. This has been illustrated earlier by perturbing the feedback gain. It can be seen from the figure that LQ design reduces the crossfeed gain by large amount.



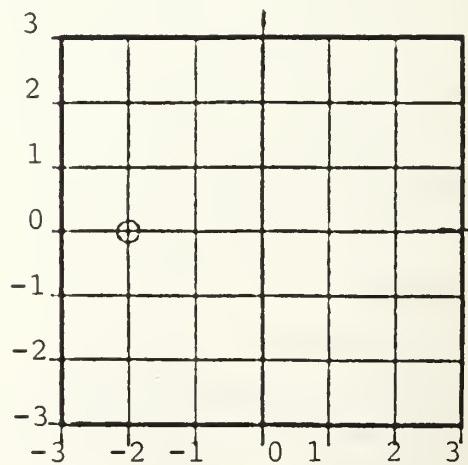
POLES (ALL)



ZERO (1 TO 1)

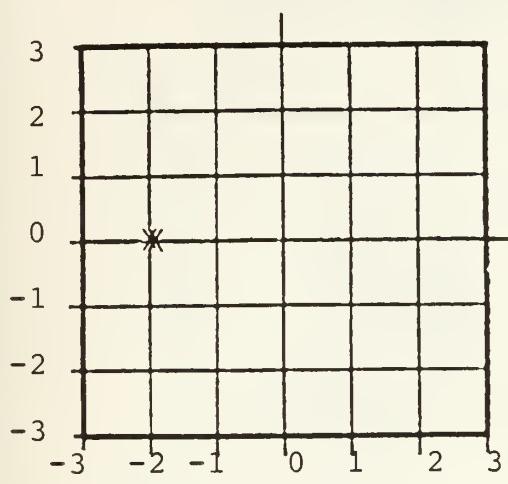


ZERO (2 TO 1)

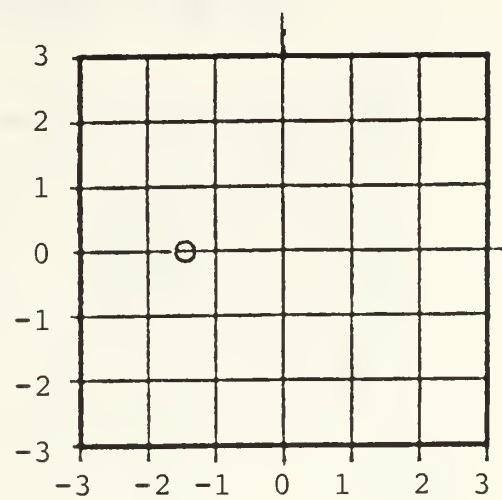


ZERO (2 TO 2)

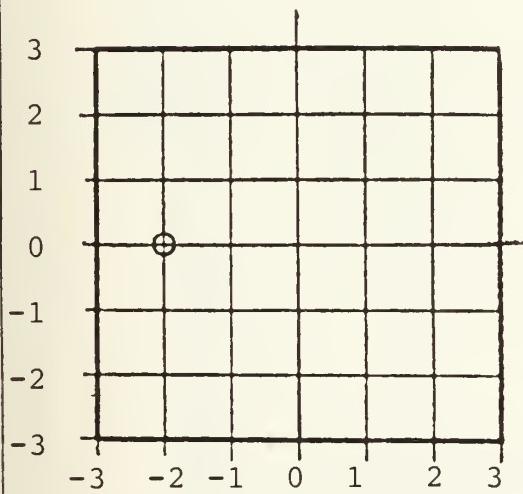
Figure 5.7 Closed-Loop Pole-Zero Plots (Unity Gain FB).



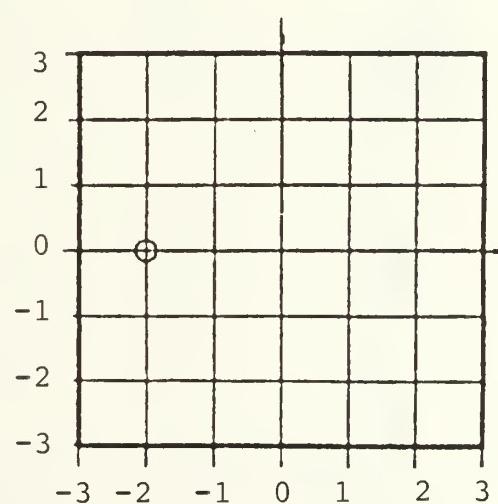
POLES (ALL)



ZERO (1 TO 1)



ZERO (2 TO 1)



ZERO (2 TO 2)

Figure 5.8 Closed-Loop Pole-Zero Plots (LQ Design).

OPEN LOOP GAIN 1 - 1

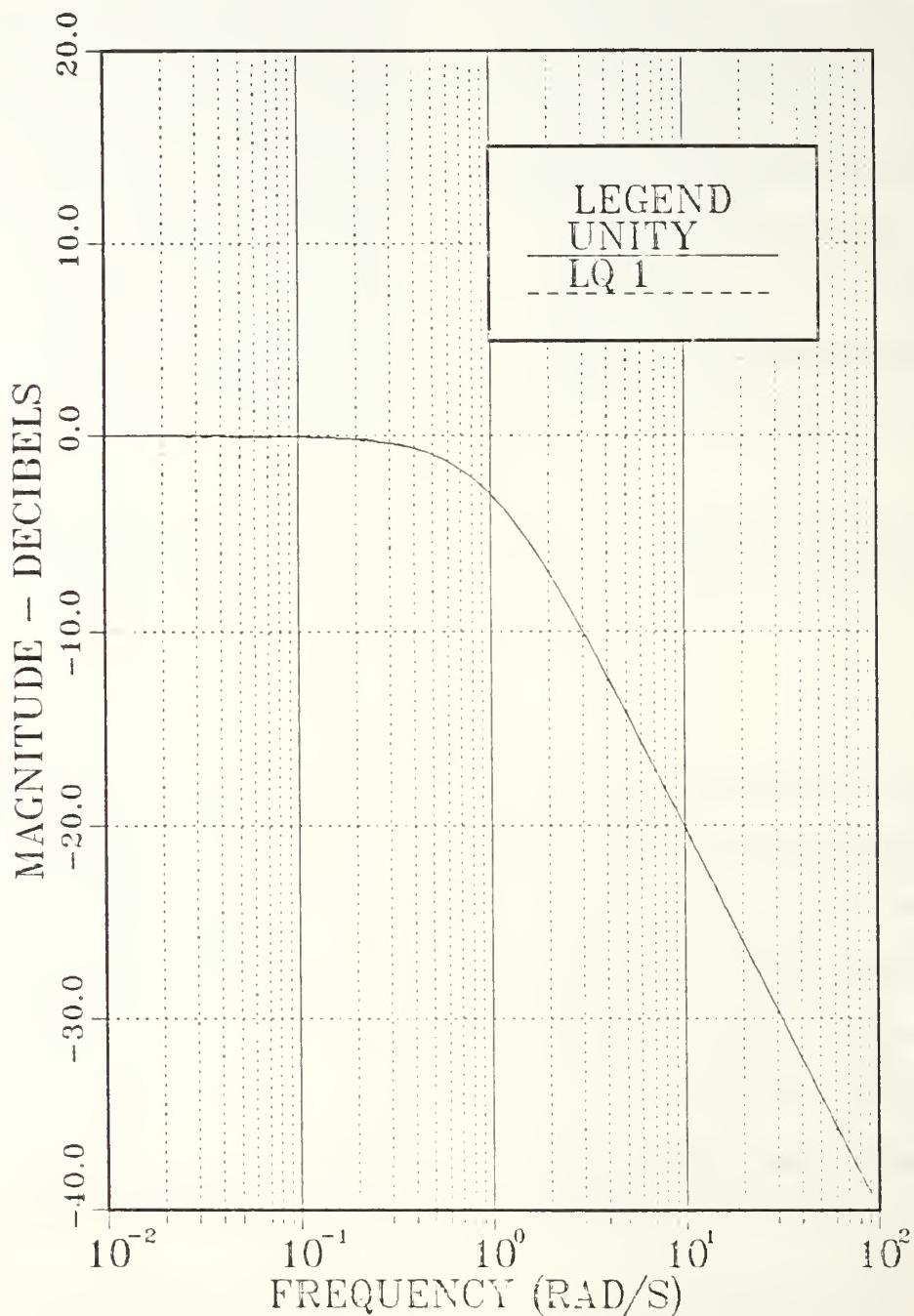


Figure 5.9 Open-Loop Bode Plots Channel 1-1.

OPEN LOOP GAIN 2 - 1

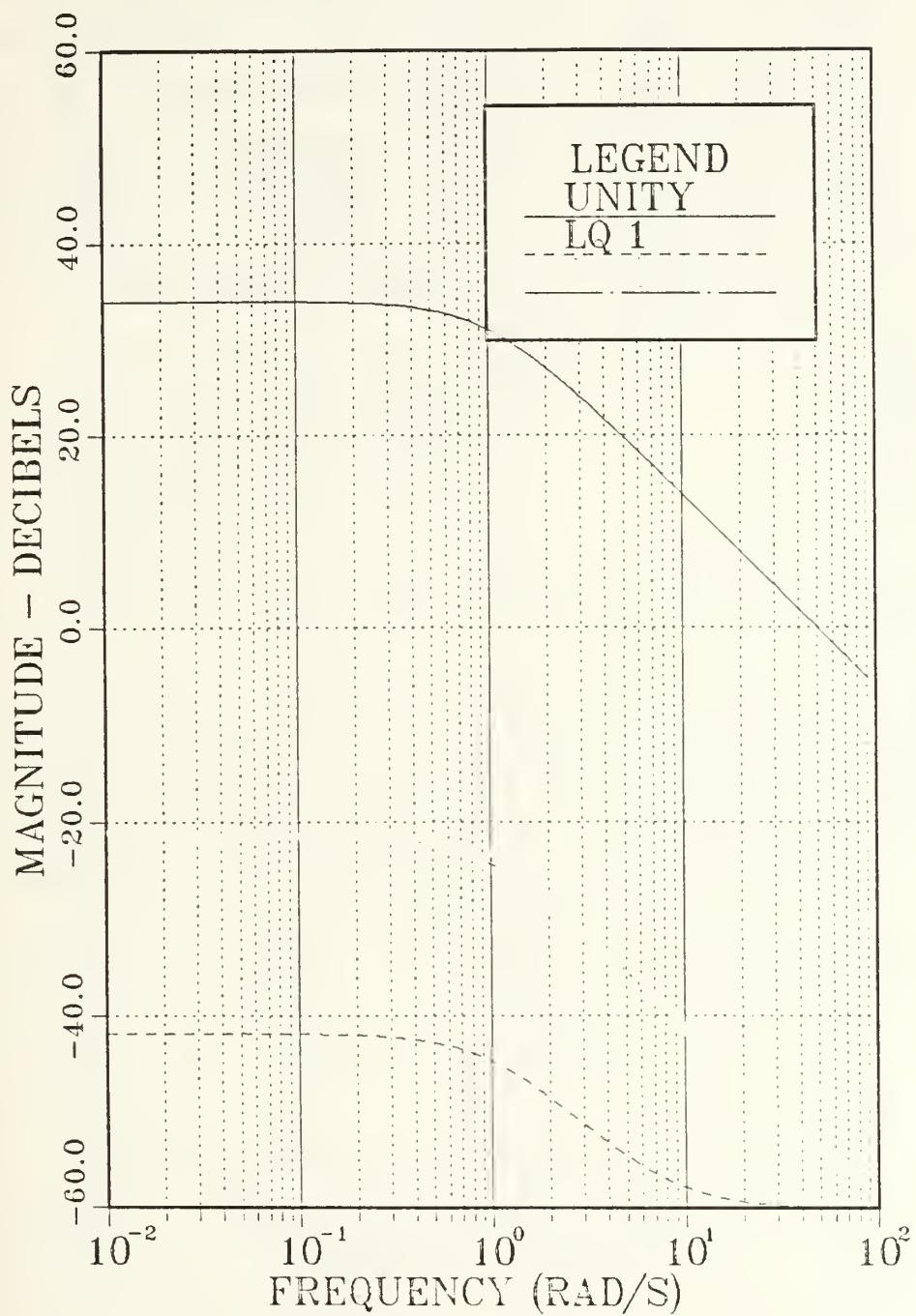


Figure 5.10 Open-Loop Bode Plots Channel 2-1.

OPEN LOOP GAIN 2 - 2

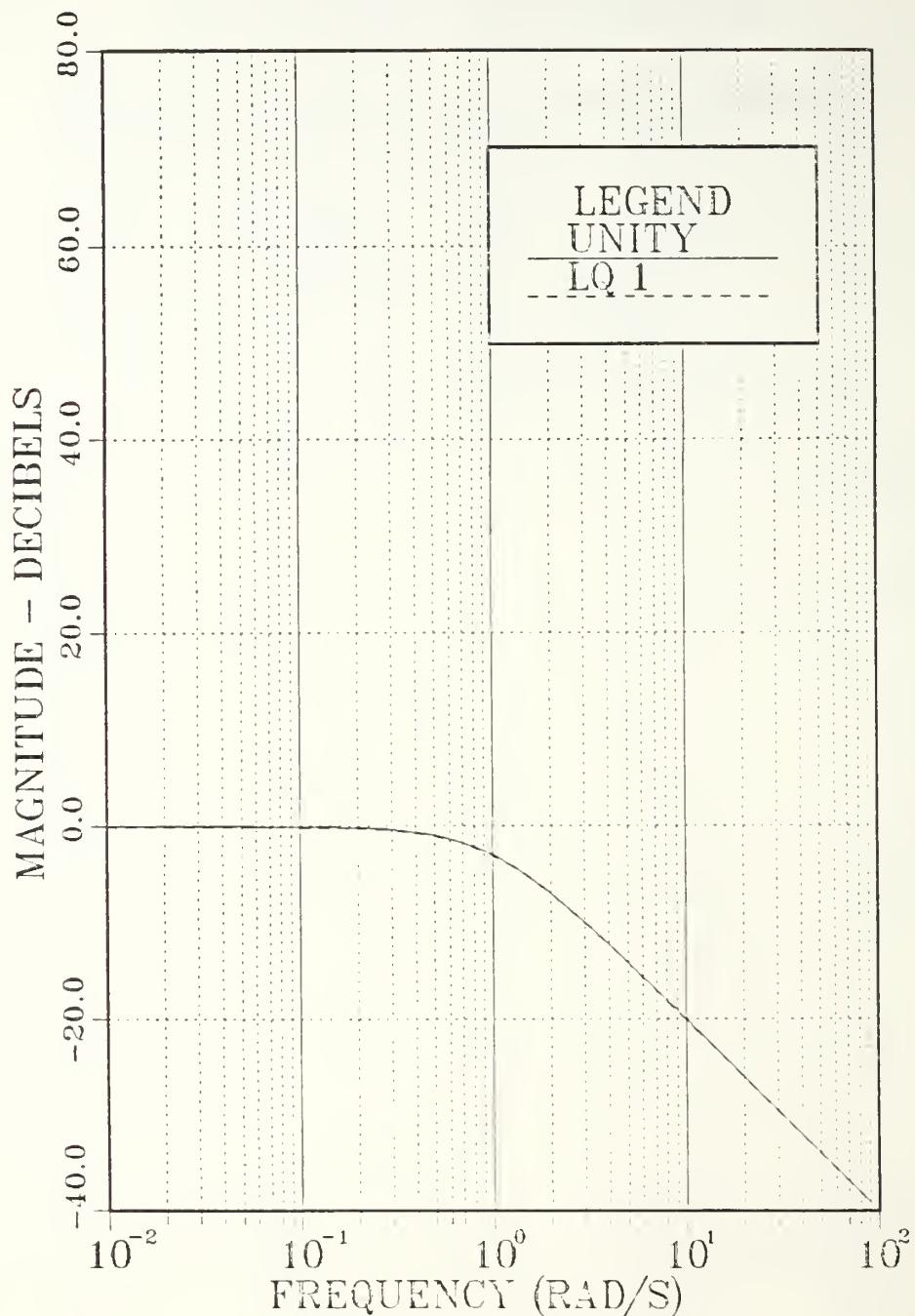


Figure 5.11 Open-Loop Bode Plots Channel 2-2.

B. A HELICOPTER DESIGN PROBLEM

The design procedure described in Chapter 4 is further illustrated in this section by an actual design problem. A controller is designed using the Linear Quadratic Pole Placement approach for the lateral dynamic model of a CH-47 helicopter. The resulting design is then compared to multi-variable state feedback controller given in [Ref. 23].

The highly coupled two inputs lateral axis model of the CH-47 helicopter is used as a full order system. The state vector $x(t)$ and control input vector $u(t)$ are given by,

$$x_1 = v = Y\text{-axis velocity (ft/sec)}$$

$$x_2 = p = \text{Roll rate (rad/sec)}$$

$$x_3 = r = \text{Yaw rate (rad/sec)}$$

$$x_4 = \phi = \text{Roll angle (rad)}$$

$$u_1 = \delta_b = \text{Yaw rate rotor deflection control (inches.)}$$

$$u_2 = \delta_c = \text{Roll rate rotor deflection control (inches.)}$$

The state variables and the body axes of the aircraft are illustrated in Figure 5.12. The yaw and roll rotor deflection control produce changes in the yaw rate, slide slip angle, roll rate and bank angle. Assuming full state feedback, the A, B, and C system matrices are given by,

$$A = \begin{bmatrix} -2.27 & 1.420 & -0.15 & 31.99 \\ .01 & -0.7 & -0.07 & 0.0 \\ 0.04 & -0.05 & -0.05 & 0. \\ 0. & 1. & 0.11 & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} 0.12 & 0.95 \\ 0.04 & -8.37 \\ .34 & .020 \\ 0.0 & .0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

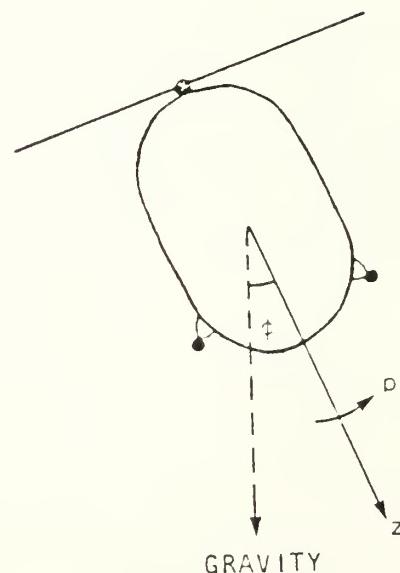
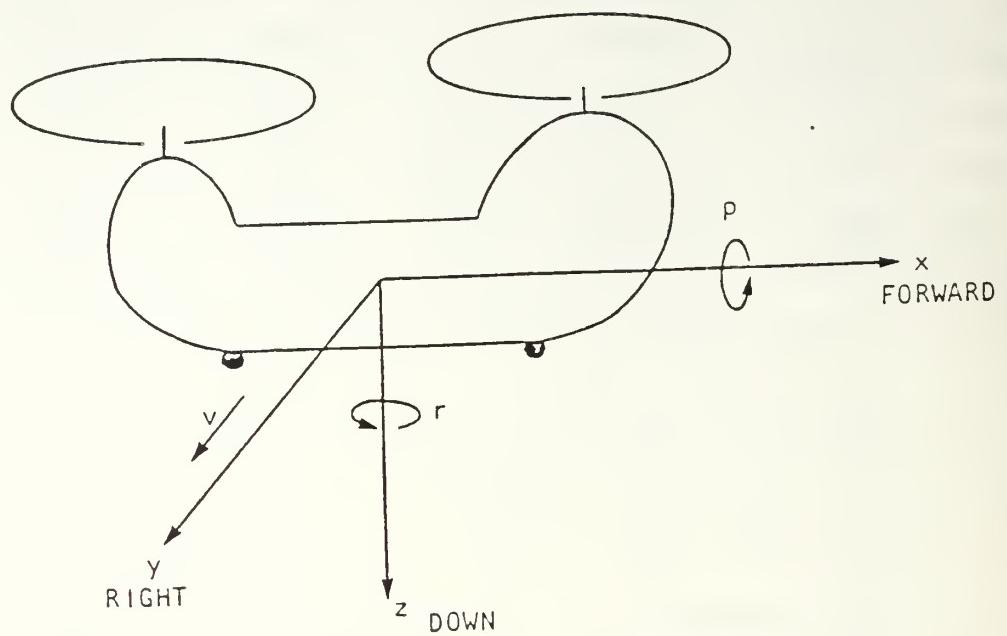


Figure 5.12 State Variables and Body Axes for CH-47.

The open loop eigenvalues of this system are;

$$\begin{aligned}s_1 &= +0.2065 \\s_2 &= -0.0503 \\s_3 &= -1.0498 \\s_4 &= -2.1263\end{aligned}$$

The open-loop system is not stable and the time response is shown in Figure 5.13 for zero input and an initial condition of $\phi(0) = 0.1$ rad

The design requirements are to satisfy specification in terms of the step input response of the roll attitude channel (ϕ/ϕ_C). Stability margin requirements are those given by standard military specification. In [Ref. 23] three designs were obtained to satisfy the desired time response performance specification. All three control laws are of the form given by,

$$u(t) = -Fx(t) + h\phi_C(t) \quad (\text{eqn 5.7})$$

The values of F and h are summarized in Table III. It was shown in [Ref. 23] that two of the designs (design 1 and 2) were extremely sensitive to model errors and perturbations. It is now shown that LQ formulation using the pole placement procedure developed here will result in robustness design and yet satisfy the conventional time response criteria.

It is assumed that the closed-loop poles requirement are the same as those obtained in the AlphaTech's design 1. (-25.12, -12.51, -9.652, -2.125). The first step in the design is to establish some asymptotic properties of the system. The dimension of the control input (2) is less than the dimension of the state(4). As shown in Chapter 2, this implies that at least two closed-loop poles approach minus infinity in the complex plane when $R \rightarrow 0$. When $R \rightarrow 0$ (meaning no control input allowed), the closed-loop poles

OPEN-LOOP RESPONSE
INITIAL ROLL ANGLE = 0.1 RAD

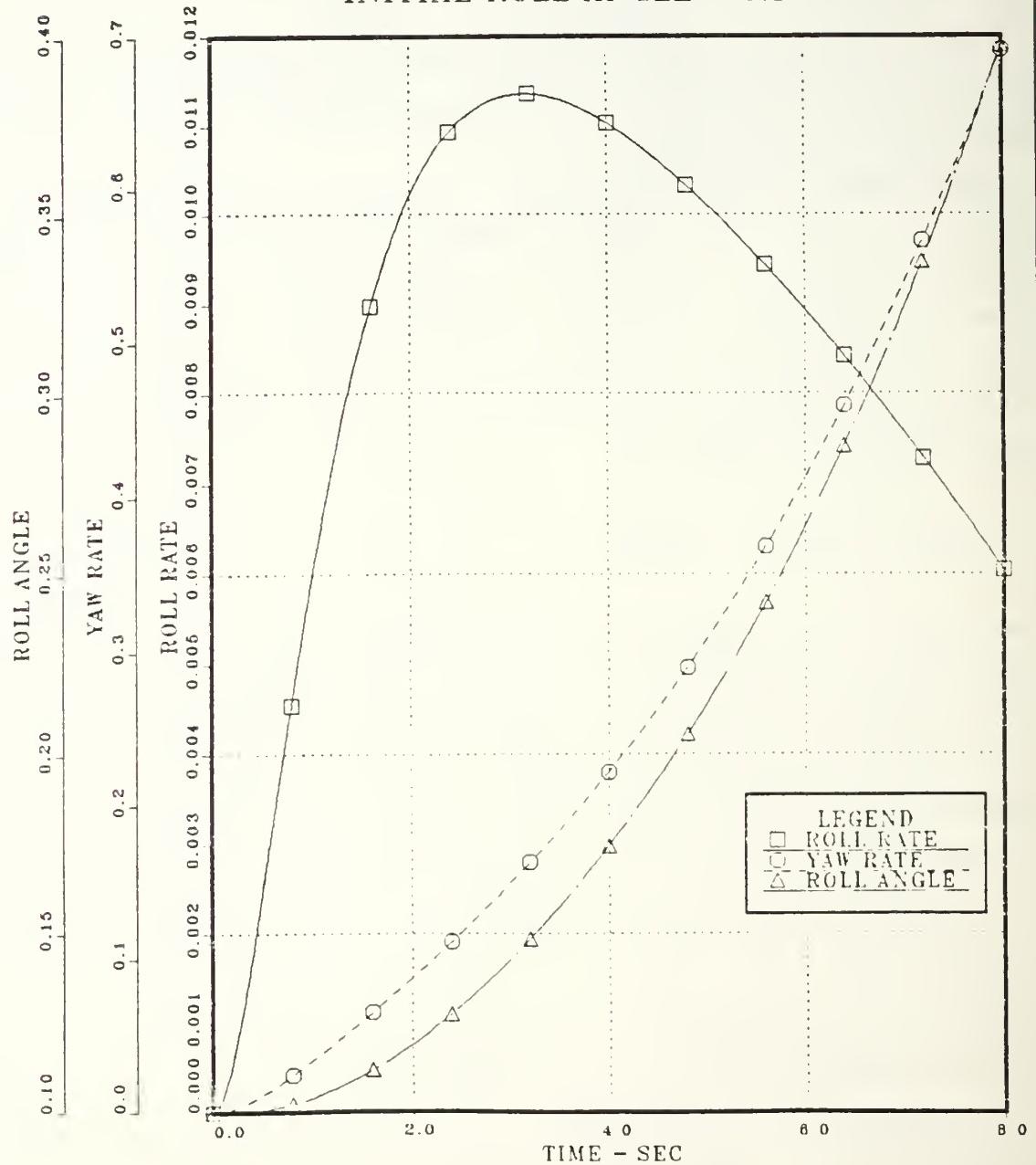


Figure 5.13 Open Loop Time Response $\phi(0) = 0.1$ rad.

TABLE III
ALPHATECH DESIGNS (1, 2, AND 3)

Designs	Feedback Gains (F)				h
one	-1.72 0.024	-23.5 -2.71	70.6 0.368	595 -7.99	595 -7.99
two	0.198 -0.01	154 -1.592	18.3 -0.189	142.0 -1.47	142.0 -1.47
three	0 0	0 -4	25.5 0	0 -27	0 -27

approach the open-loop poles or their mirror images if they are in the right hand plane. (i.e. -0.2065, -0.0503, -1.04987, and -2.12635).

The next step in the design procedure is to select a suitable starting control weighting matrix. Based on guideline given in [Ref. 4] and assuming that $u_{1\max}$ and $u_{2\max}$ are equal to 1 inch. $R = I$ is selected for the initial design. (In the present formulation ρ is not needed as only Q is varied to place the pole).

The sequence of the reassignment is determined next. At present there is no known established guideline for selecting the preferred sequence of moving the poles. Two extreme sequences are considered as follows,

A. Move the left most open-loop pole to the left most desired closed-loop pole etc. The reassignment sequence then become:

1. move pole at -2.1265 to -25.12
2. move pole at -1.0483 to -12.51

3. move pole at -0.2065 to -9.652
4. move pole at -0.0503 to -2.125

B. Move the right most open-loop pole to the left most desired closed-loop pole etc. For the problem given, the reassignment sequence are :

1. move pole at +0.2065 to -25.12
2. move pole at -0.0503 to -12.51
3. move pole at -1.0498 to -9.652
4. move pole at -2.126 to -2.125
(close, so no move required)

Using the pole placement program developed in this thesis, the corresponding Q and F for each of the reassessments are obtained and tabulated in Tables IV and V for the two cases selected (design LQ-A and LQ-B). It can be seen from the table that difference reassignment sequence results in different set of Q_e and F_e . In general, once the matrix R and the n closed-loop poles are selected, there are $n(n+1)/2$ extra degrees of freedom available in Q. This also verifies the well-known fact that the general solutions of Q and R for a given set of closed-loop eigenvalue are non-unique. The elements of the Q matrix obtained during each reassignment depends on the transformation matrix (M, L or U) and hence the eigenvectors that are used to construct them. The important of eigenvector type of assignment is evident as Q, F and the resulting closed-loop poles depend on the eigenvectors used. The designer can shape the design by choosing appropriate eigenvectors for the transformation matrix. The application of these extra degrees of freedom will be discussed in the following section. The resulting designs (LQ-A and LQ-B) are now compared with the first design in [Ref. 23].

TABLE IV
RESULTS FROM POLE PLACEMENT SEQUENCE(LQ-A)

Move	Q and F obtained during each reassignment			
Q_1	0.10027	0.96194	0.11959	-1.50849
	0.96194	9.22847	1.14725	-14.47182
	0.11959	1.14725	0.14262	-1.79908
	-1.50849	-14.47182	-1.79908	22.69429
F_1	0.00387	0.03151	0.00399	-0.06416
	-0.24260	-2.83522	-0.34541	3.11642
Q_2	6.12853	-2.99198	0.51376	-95.14285
	9.198	1.46070	-0.25082	46.44916
	0.51376	-0.25082	0.04307	-7.97589
	-95.14284	46.44916	-7.97589	1477.05225
F_2	0.05825	-0.02844	0.00490	-0.90426
	2.27529	-1.11079	0.19079	-35.32260
Q_3	80.37563	18.91502	13.64762	218.27522
	18.91502	4.45133	3.21173	51.36732
	13.64761	3.21173	2.31734	37.06267
	218.27521	51.36732	37.06267	592.76758
F_3	1.58772	0.37365	0.26961	4.31182
	-8.62241	-2.02917	-1.46414	-23.41618
Q_4	0.15575	0.02839	-2.48989	0.19453
	0.02839	0.00517	-0.45383	0.03546
	-2.48989	-0.45383	39.80322	-3.10981
	.19453	0.03546	-3.10981	0.24297
F_4	-0.38007	-0.06927	6.07574	-0.47469
	-0.06411	-0.01168	1.02482	-0.08007
Q_e	86.76016	16.91336	1.79108	121.81841
	16.91336	15.14567	3.65433	83.38010
	11.79107	3.65433	42.30624	24.17787
	121.81841	83.38010	24.17787	2092.75684
F_e	1.26977	0.30745	6.35424	2.86871
	-6.65383	-5.98686	-0.59394	-55.70242

$$u(t) = -Fx(t) + h \phi_C(t), \quad h = \begin{bmatrix} 2.8687 \\ -55.7024 \end{bmatrix}$$

TABLE V
RESULTS FROM POLE PLACEMENT SEQUENCE(LQ-B)

Move	Q and F obtained during each reassignment			
Q_1	0.00031	0.05289	0.00589	0.04781
	0.05289	9.06491	1.00865	8.19296
	0.00589	1.00865	0.11223	0.91163
	0.04781	8.19296	0.91163	7.40489
F_1	0.00017	0.02851	0.00317	0.02576
	-0.01771	-3.03548	-0.33776	-2.74347
Q_2	0.26982	0.46312	16.33746	10.89106
	0.46312	0.79492	28.04224	18.69380
	16.33746	28.04222	989.23975	659.45752
	10.89106	18.69380	659.45728	439.61426
F_2	0.44607	0.76563	27.00978	18.00517
	-0.26207	-0.44982	-15.86824	-10.57837
Q_3	0.16317	-0.35205	6.24519	-9.35123
	-0.35205	0.75956	-13.47424	20.17563
	6.24519	-13.47425	239.02527	-357.90405
	-9.35123	20.17563	-357.90381	535.90698
F_3	0.19939	-0.43018	7.63129	-11.42668
	0.30220	-0.65201	11.56628	-17.31876
Q_e	0.43330	0.16396	22.58853	1.58764
	0.16396	10.61939	15.57663	47.06238
	22.58853	15.57661	1228.37695	302.46509
	1.58764	47.06238	302.46509	982.92603
F_e	0.64563	0.36396	34.64423	6.60425
	0.02242	-4.13731	-4.63971	-30.64059

$$u(t) = -F_x(t) + h \phi_c(t), \quad h = \begin{bmatrix} 6.60425 \\ -30.64059 \end{bmatrix}$$

Closed loop time response for a step input in roll attitude command ($\phi_c = 0.1$ rad) for the three designs are shown in Figures 5.14 to 5.16. It can be seen from the response plots that although all three designs have almost identical closed-loop pole location, the step response for various states differs. The AlphaTech design's response is characterized by the large 'overshoot' in the yaw rate step response. In the Linear Quadratic Design (LQ-A), the 'overshoot' occurs with the roll attitude response rather than

the yaw rate response; the yaw rate response is well damped. The LQ-B design appears to be a better design as coupling among modes are small and can not be detected from the time response plot. It is noted that the difference in response for the same set of closed-loop eigenvalue is due to coupling among various modes through their respective eigenvectors. The 'overshoot' in this case is obviously not due to complex conjugate poles as all closed-loop poles in the three designs are on the real axis. The set of final eigenvector for the three designs are tabulated in Table VI. Inter-mode coupling for the yaw rate response in the AlphaTech design and roll attitude response in the LQ design case A can be readily seen from the table. The issue of eigenvector assignment will be discussed in the next section.

TABLE VI
CLOSED-LOOP EIGENVECTORS

Eigenvalues/Eigenvector				
Alpha one	-24.79907	-2.12242	-9.72403	-11.9449
	-0.39336	-0.99976	-0.48016	-0.41854
	0.10767	0.00822	-0.32948	-0.33640
	-0.91306	0.01955	-0.81181	-0.84283
LQ-A	-0.00029	-0.00488	0.04306	0.03593
	-2.12499	-9.26645	-12.52066	-25.20990
	0.22870	-0.15459	0.01894	-0.12653
	0.18018	-0.98218	-0.99664	0.99118
LQ-B	-0.95603	-0.01258	-0.00396	0.00031
	-0.03529	0.10614	0.07964	-0.03932
	-25.17894	-9.62708	-12.51644	-2.12630
	-0.12652	-0.13209	0.05042	-0.99977
	0.99118	-0.98557	-0.99375	0.00797
	0.00024	-0.02568	0.06075	0.01945
	-0.03937	0.10267	0.07886	-0.00475

CLOSED-LOOP RESPONSE ALPHATEC 1
STEP INPUT IN ROLL ANGLE 0.1 RAD

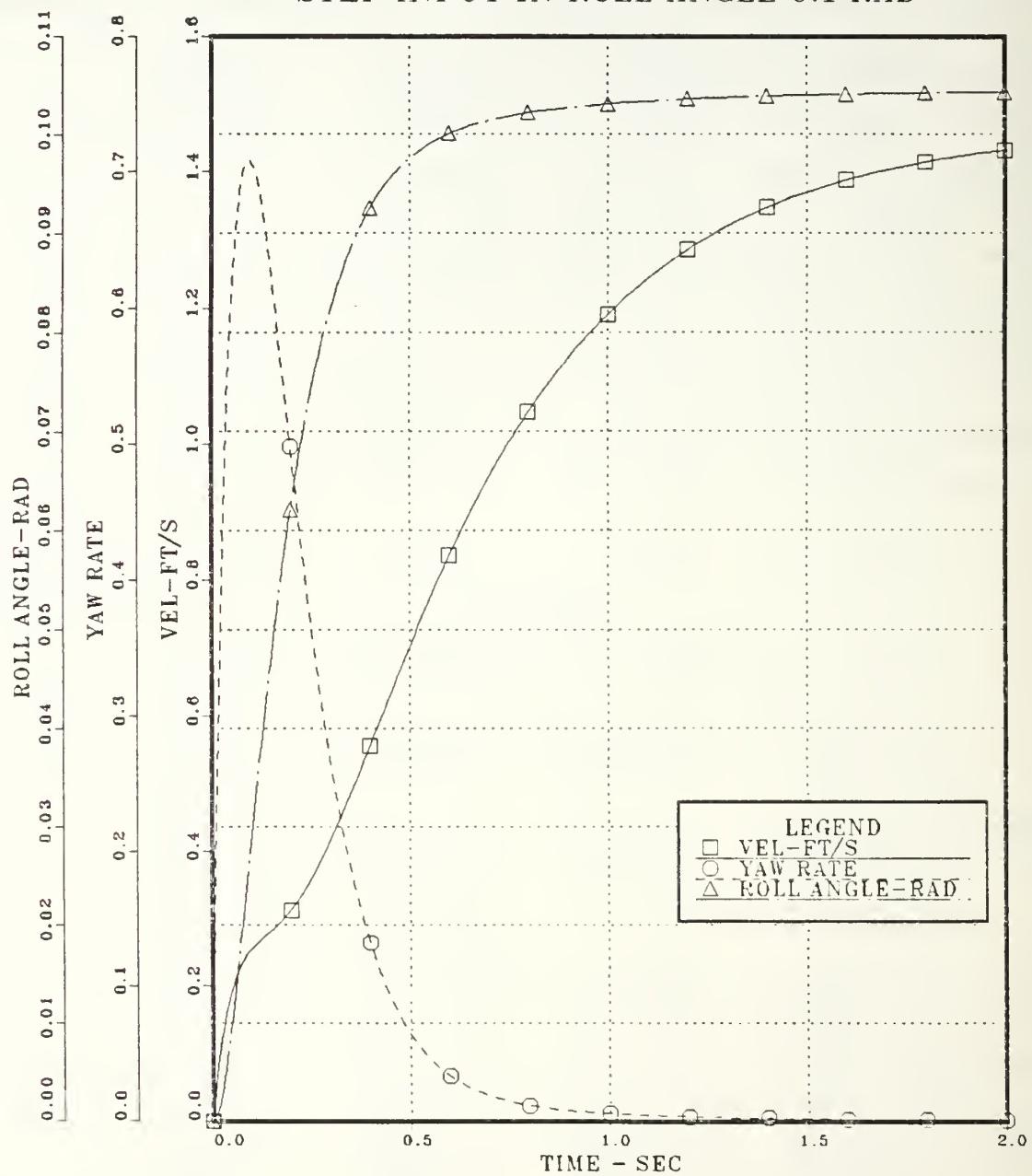


Figure 5.14 Closed-Loop Time Response Plot (AlphaTech 1).

CLOSED LOOP RESPONSE LQ-A
STEP INPUT IN ROLL ANGLE 0.1 RAD

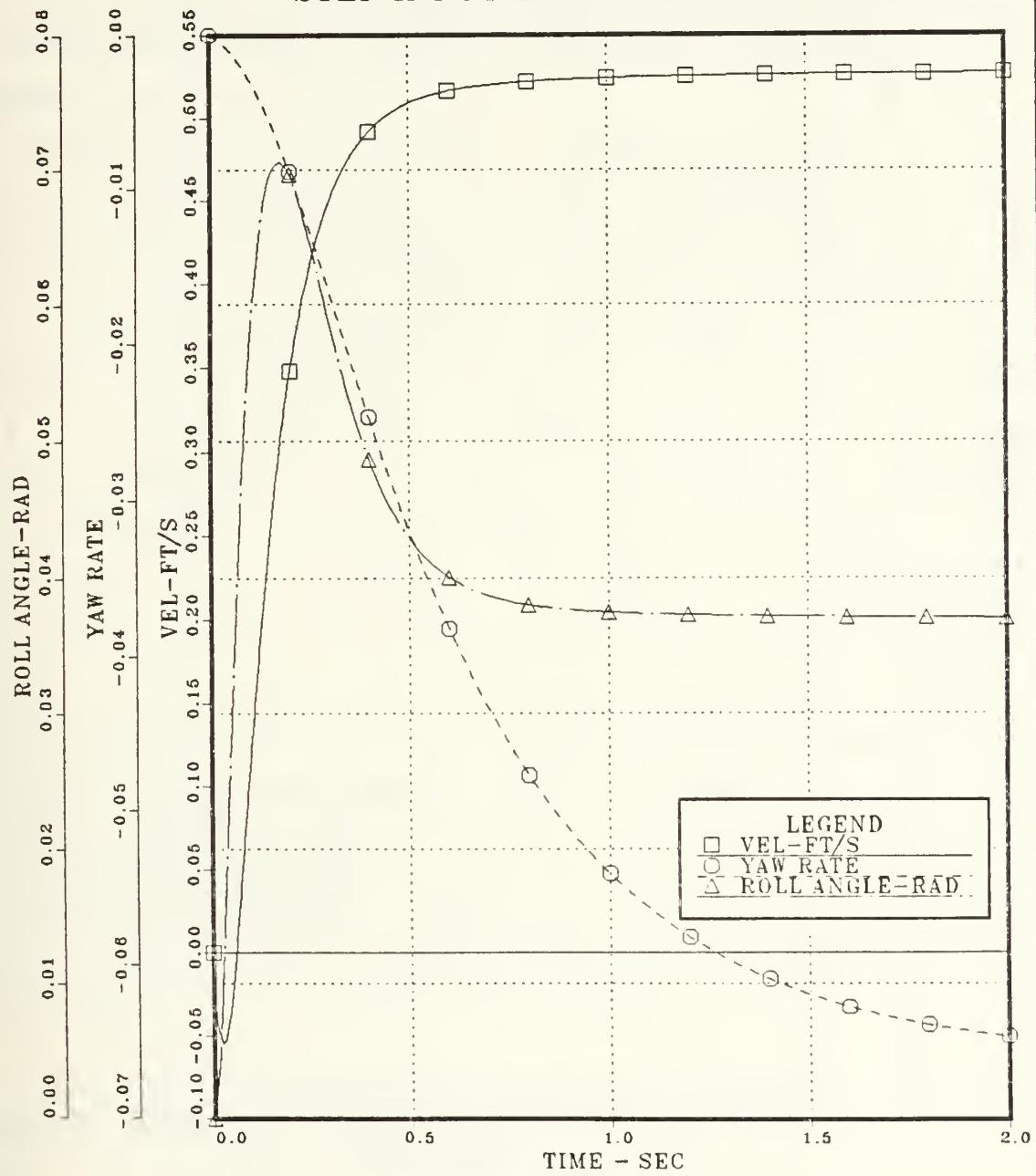


Figure 5.15 Closed-Loop Time Response Plot (LQ-A).

CLOSED LOOP RESPONSE LQ-B
STEP INPUT IN ROLL 0.1 RAD

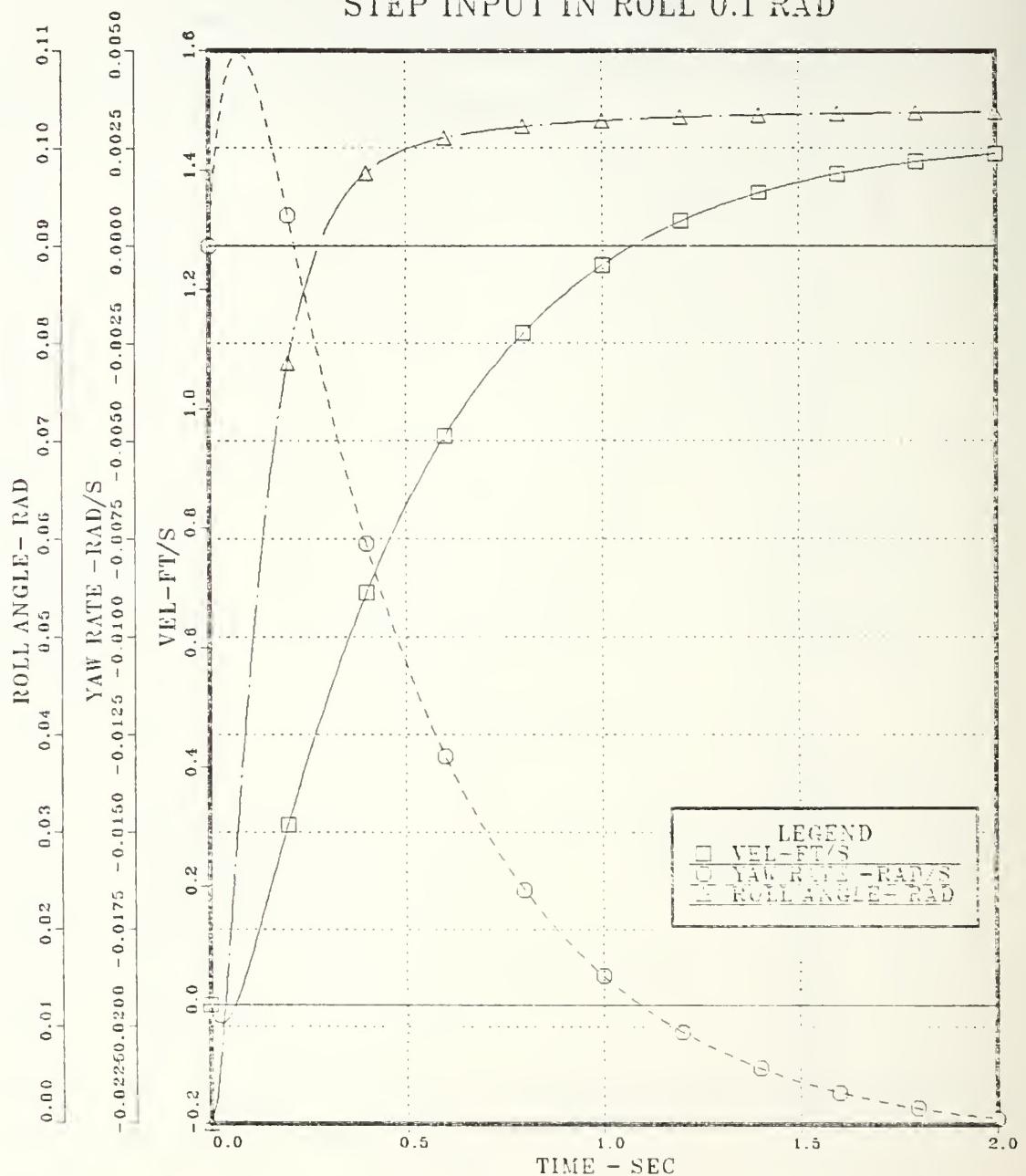
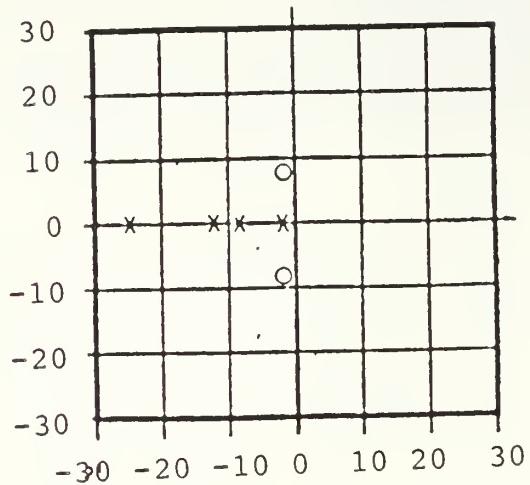


Figure 5.16 Closed-Loop Time Response Plot (LQ-B).

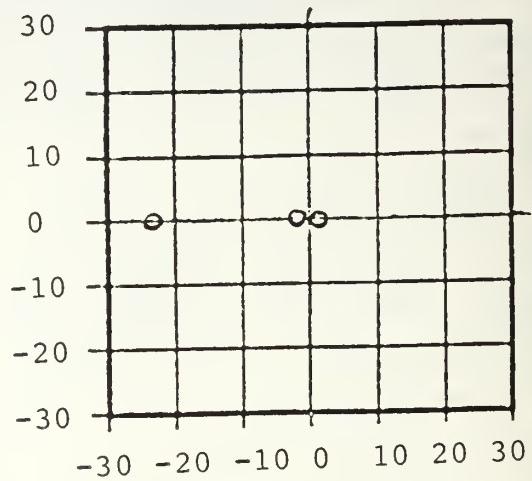
The difference in time response performance for the three designs can also be interpreted in term of the closed-loop pole-zero plots in the frequency domain. The pole-zero plots for the designs, with ϕ_c as the input, are shown in Figures 5.17 to 5.19 . As all pole locations are the same for the three designs, they are indicated only once in each diagram.

In the AlphaTech's design, the zero locations for the various channels clearly indicate the weakness in the design. Most transmissions zeros are located away from the poles locations. The system is therefore strongly coupled to its external environment. A good example is the lightly "damped" pair of zero at $(-1.05, \pm 7.31j)$ for the $v - \phi_c$ channels. It is in fact these undesired zeros that reduce the overall robustness of the system. The mechanism of robustness improvement in the LQ design can also be seen in the pole-zero plots in Figures 5.18 and 5.19 . The built-in robustness in the LQ design causes the zeros at various channels to move to locations where their transmission properties can be canceled by the closed-loop poles. An excellent example is in the $v - \phi_c$ channel, where the zeros at -2, -7, and the mirror image of +24.0 are fairly close to the closed-loop pole locations at -2.12, -9.26 and -25.12. In the case where the zero from LQ design are not close to the closed-loop poles ($r - \phi_c$ channel for LQ-A) the zeros are "well damped" and their transmission properties can be neglected.

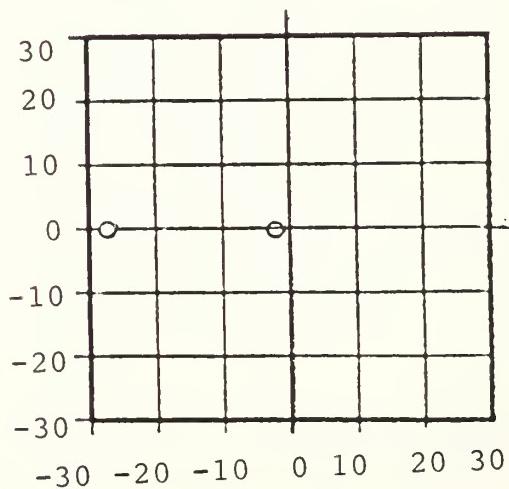
Robustness properties of the three designs presented here can also be analyzed from the open-loop Bode plots. The open-loop transfer function gain plots for channel 1-1, 1-2, 2-1 and 2-2 for the three designs are shown in Figures 5.20 to 5.24 . Cross coupling problem for the AlphaTech design is clearly indicated by the relatively high gain of



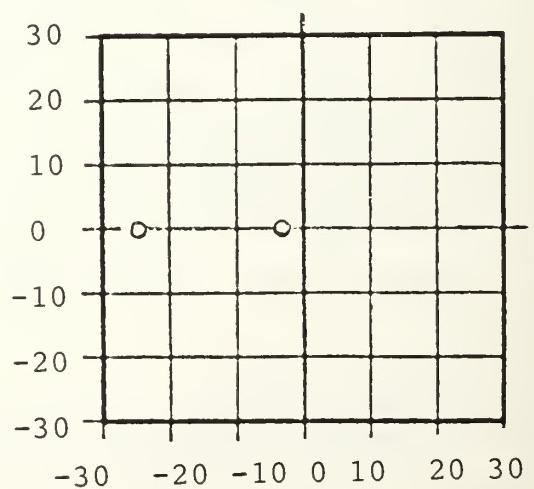
$v - \phi_c$



$\dot{\phi} - \phi_c$

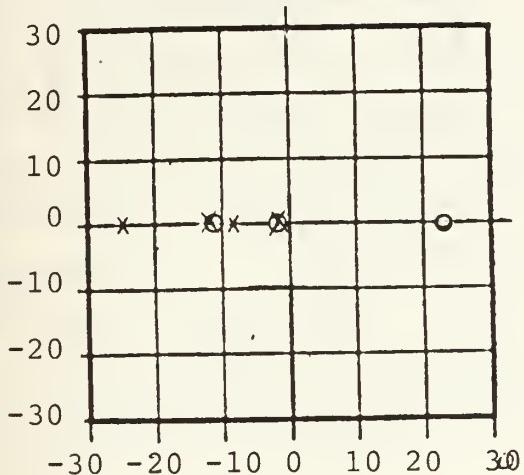


$R - \phi_c$

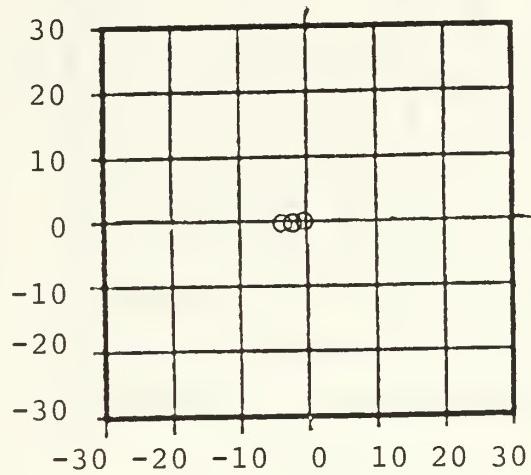


$\dot{\phi} - \phi_c$

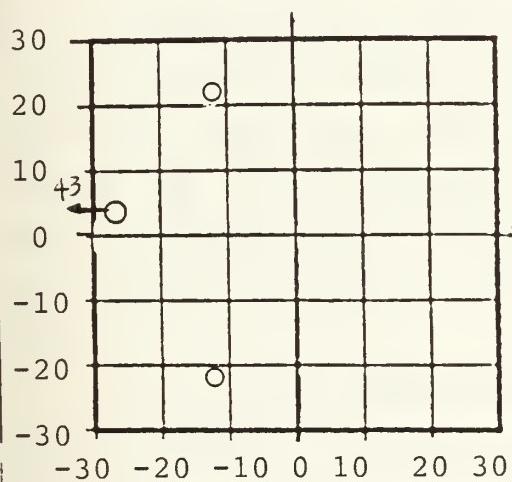
Figure 5.17 Closed-Loop Pole-Zero Plots (AlphaTech 1).



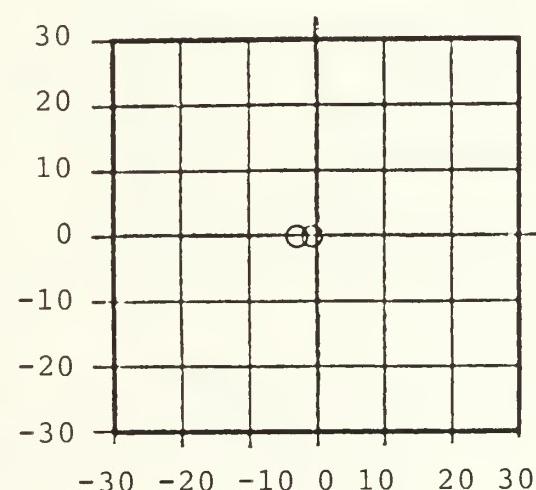
$v - \phi_c$



$\dot{\phi} - \phi_c$

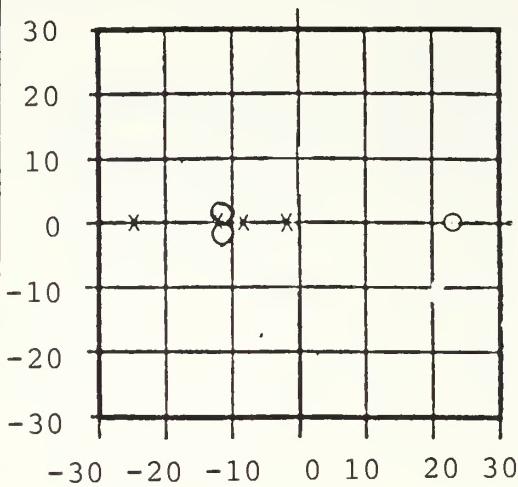


$R - \phi_c$

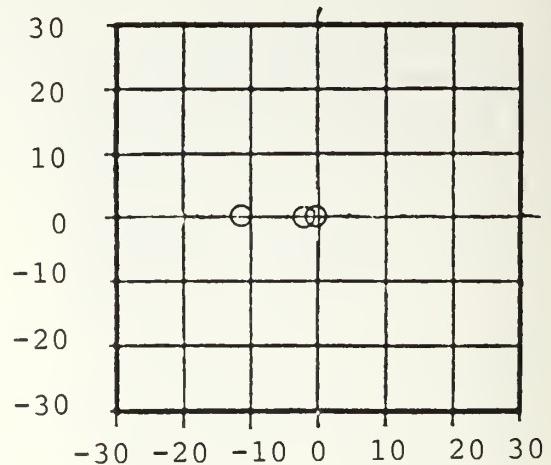


$\phi - \phi_c$

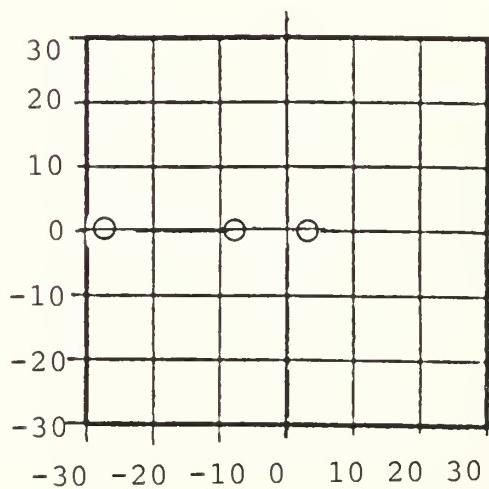
Figure 5.18 Closed-Loop Pole-Zero Plots (LQ-A).



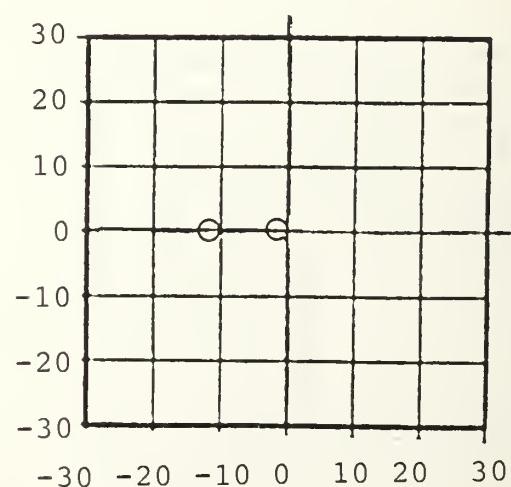
$$v - \phi_c$$



$$\dot{\phi} - \phi_c$$



$$R - \phi_c$$



$$\phi - \phi_c$$

Figure 5.19 Closed-Loop Pole-Zero Plots (LQ-B).

the channel from input 2 to input 1 (2-1). Both LQ designs presented here reduced the gain in this channel by more than 40 dB in the frequencies of interest. The bandwidth is reduced from above 100 rad/s to about 8 rad/s. Unlike the simple (2x2) system where there was little difference between the direct channels of the two designs, gain adjustment is observed in all channels. For example, LQ designs reduce the gain in channel 1-1 but increase the gain in channel 2-2. There is also a slight increase in the coupling channel 1-2. The overall effect is that of gain balancing, gains in channels that are affected by cross-coupling perturbation are lowered together with some adjustments in other channels. Different reassignment sequence results in different adjustments. The designer has to choose a set of gain curve depending on the particular requirement. The relationship between the open-loop Bode plots and the zeros of the system is also evident from these diagrams. The low frequency resonance (or 'peak') in the gain vs frequency plot for the AlphaTech design correspond to the undesirable zero mentioned earlier. In the LQ designs, these low frequency resonances are absent because of the more desirable zero locations. It is also interesting to note that the LQ-B gain curves fall nicely in between the other two designs. This fact, together with its better overall low frequency gain characteristic, may account for the better time response behavior of the LQ-B design.

As a final comparison, the closed-loop frequency response plots of the various channel for the three designs are shown in Figures 5.25 to 5.32 . At high frequency, all the gain plots approach the -20dB/decade slope as all eigenvalues are on the real axis. The AlphaTech design has a near 0db gain for frequency up to about 2 rad/s. It appears

OPEN LOOP GAIN 1 - 1

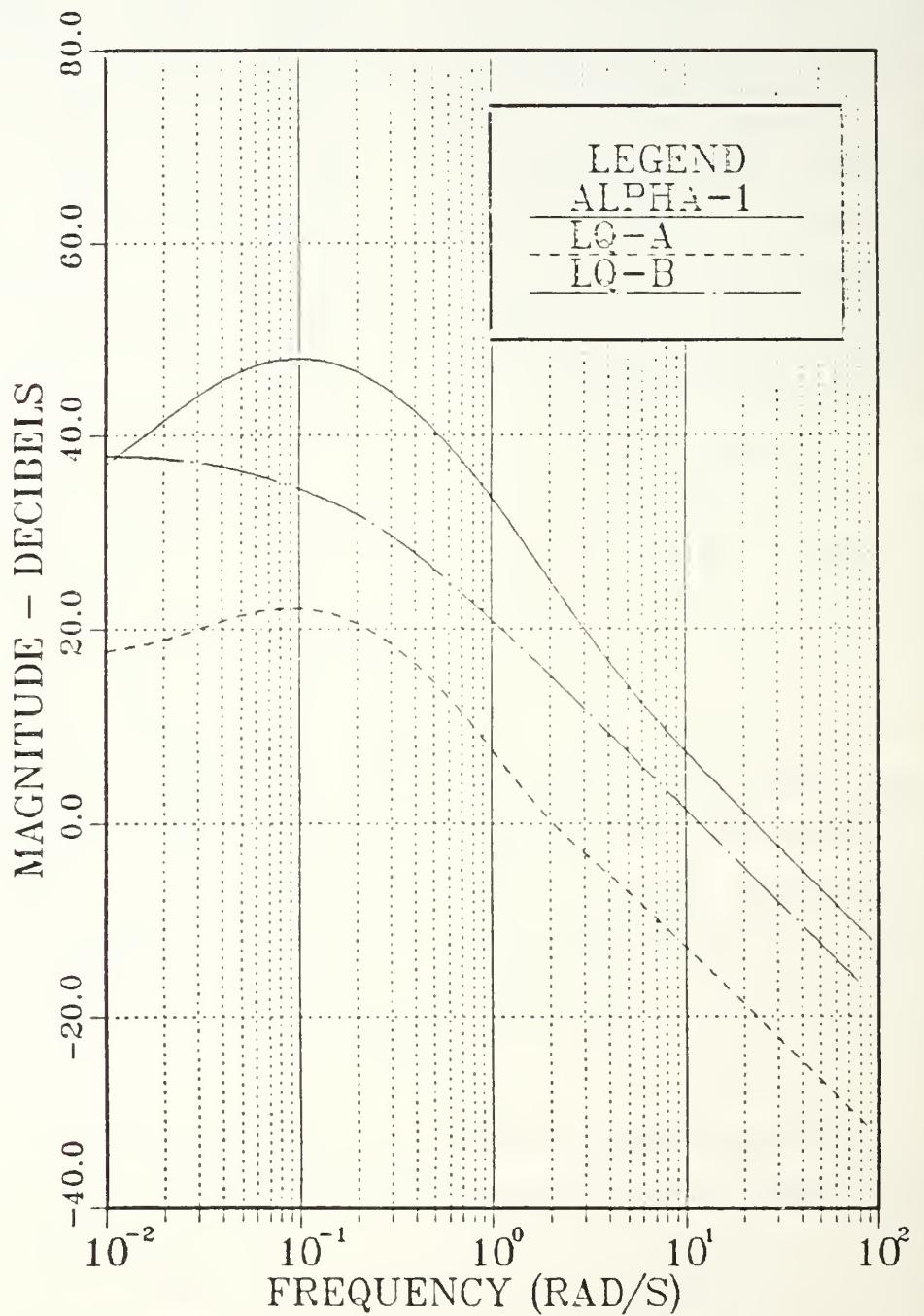


Figure 5.20 Bode Plots Comparison- Input 1-1.

OPEN LOOP GAIN 1-2

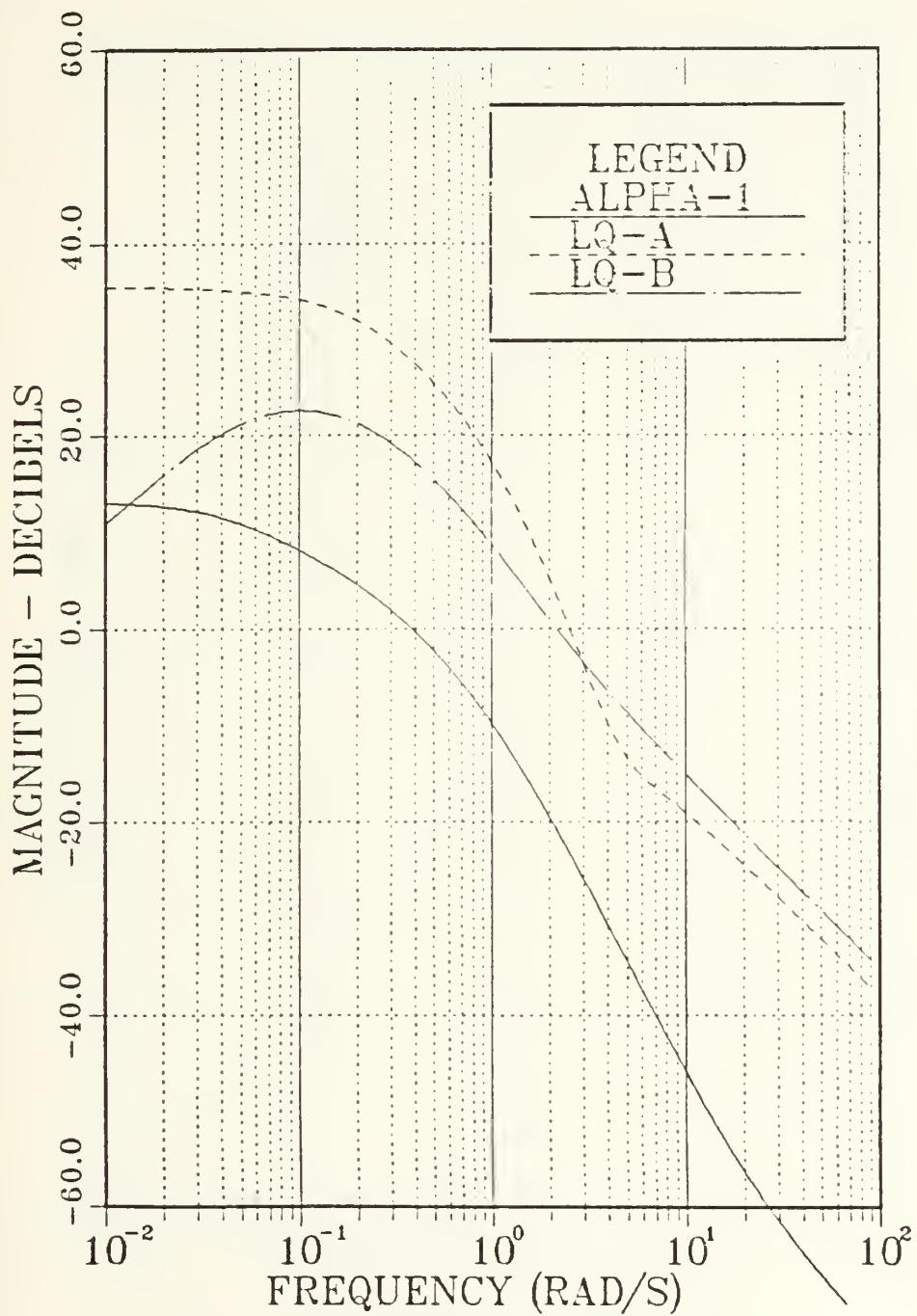


Figure 5.21 Bode Plots Comparison - Input 1-2.

OPEN LOOP GAIN 2-1

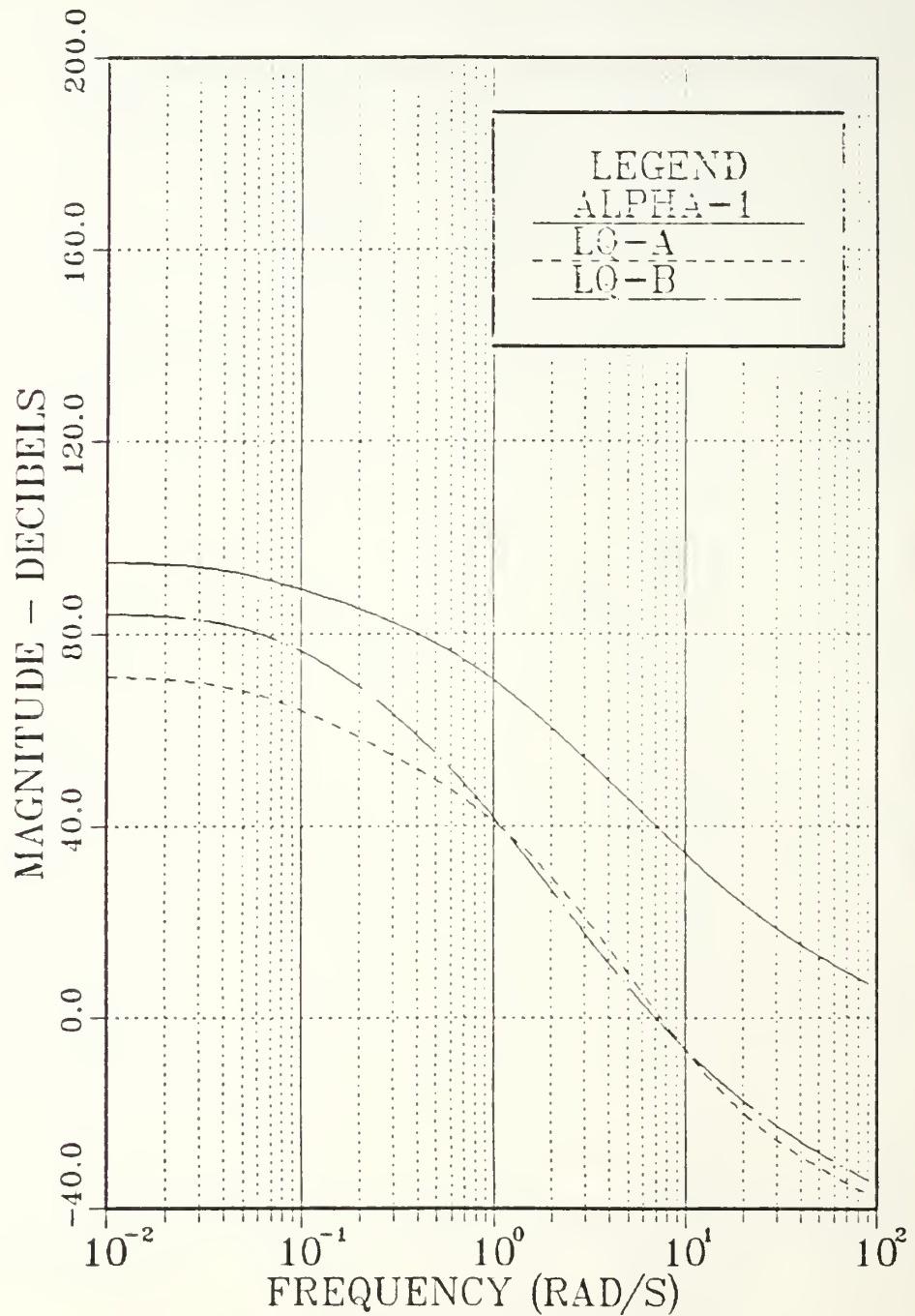


Figure 5.22 Bode Plots Comparison - Input 2-1.

OPEN LOOP GAIN 2-2

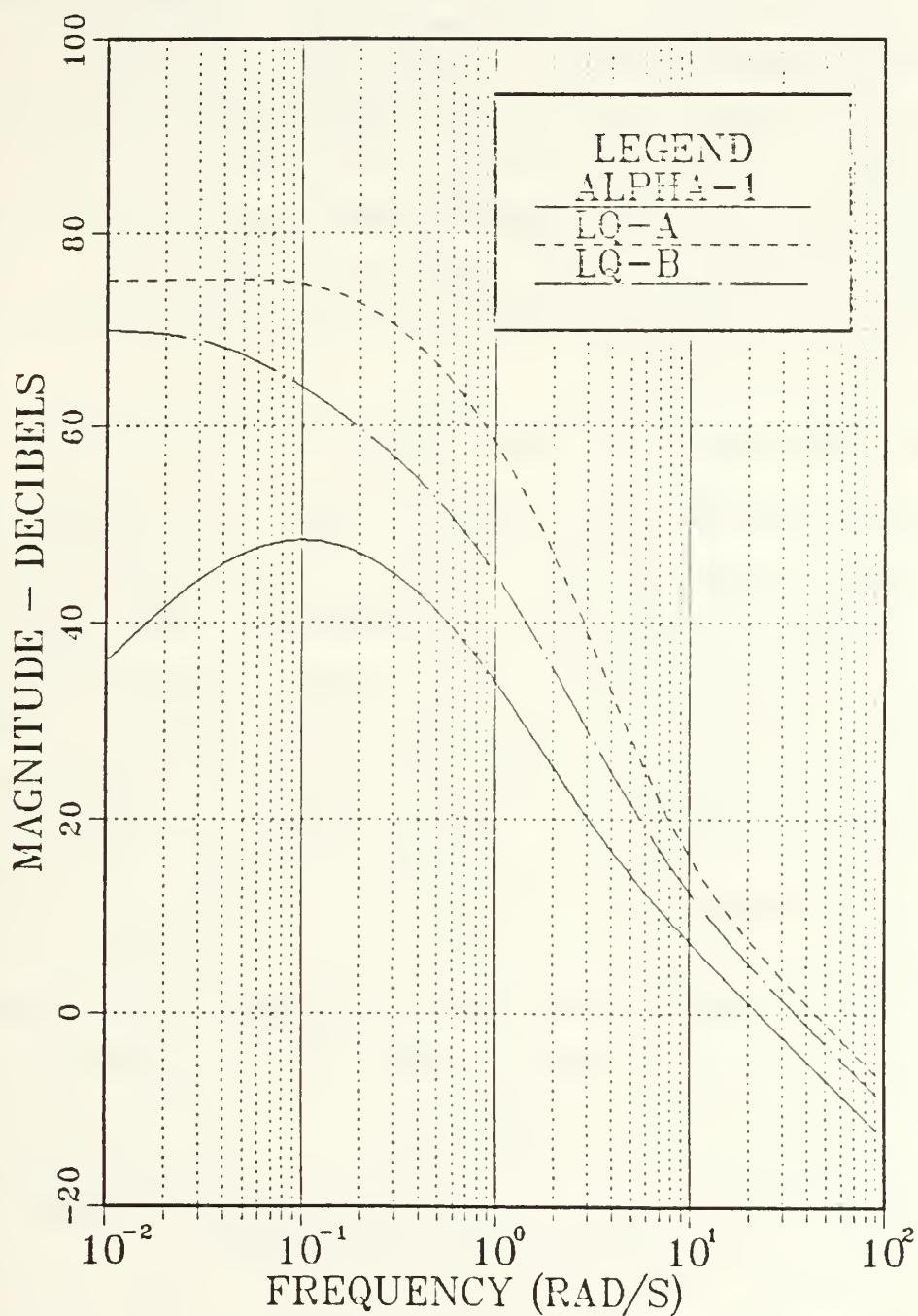


Figure 5.23 Bode Plots Comparison - Input 2-2.

to be an acceptable design but was shown to have undesired transmission zeros in other channels; this can also be seen from the closed-loop plot for the $r-\phi_C$ channel in Figure 5.31 . Design LQ-A, obtained from the reassignment sequence A, is characterized by very low DC gain (-9dB) and 'peak' in frequencies near 10 rad/s. These have been shown, both in the time response and pole-zero discussions that it will have undesired effect on the pilot's control. Design LQ-B, obtained from the second reassignment sequence appears to be the best compromise. All channels (Figures 5.30 to 5.32) have flat low frequency characteristic and coupling between modes are almost absent.

C. DISCUSSION AND CONCLUDING REMARKS

A new computer aided design procedure for the multivariable linear time-invariant system using Linear Quadratic Pole Placement formulation is presented. The two design examples presented above have served to demonstrate the complexity involved in a multivariable design. The main problem lies in the fact that the solution of a MIMO problem is in general non-unique. This was shown in the helicopter problem where different approaches result in different designs, although the closed-loop eigenvalues for all designs were the same. It was also shown that the extra degrees of freedom in MIMO system design can be accounted for by analyzing the singular value plots, transmission zero movements and closed-loop eigenvectors of the designs. It becomes apparent that the success of any multivariable design methodology hinges on how to make use of these extra degrees of freedom available in MIMO system. It was also shown in the above design examples that the Linear Quadratic Pole Placement procedures possess such quality. Some unique properties of the method presented here are as follows;

CLOSED-LOOP BODE PLOT($V - \phi_C$)

INPUT # = 1
OUTPUT # = 1
TF GAIN = $6.381 \cdot 10^1$

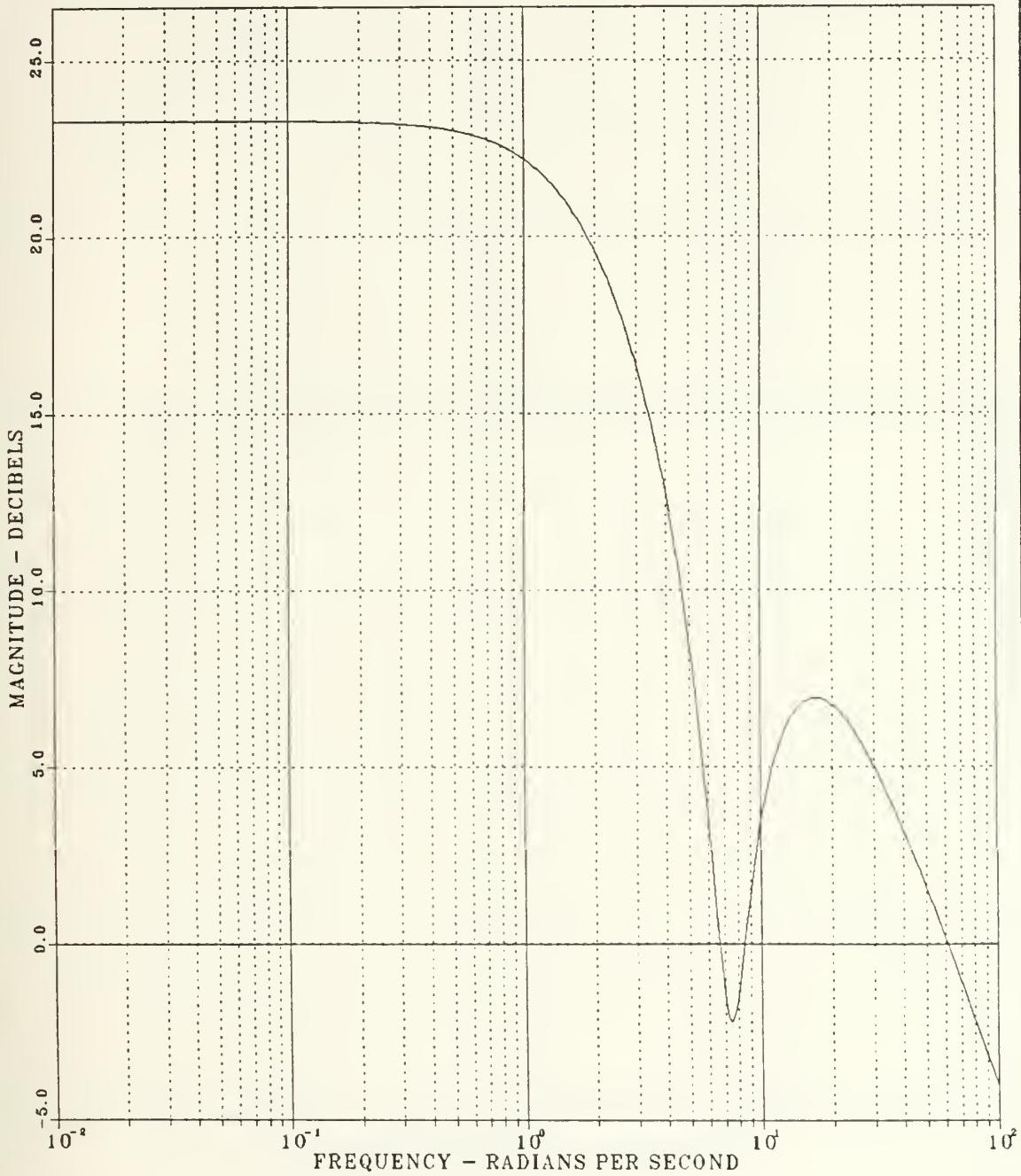


Figure 5.24 Closed-Loop Bode Plot- AlphaTech 1-1.

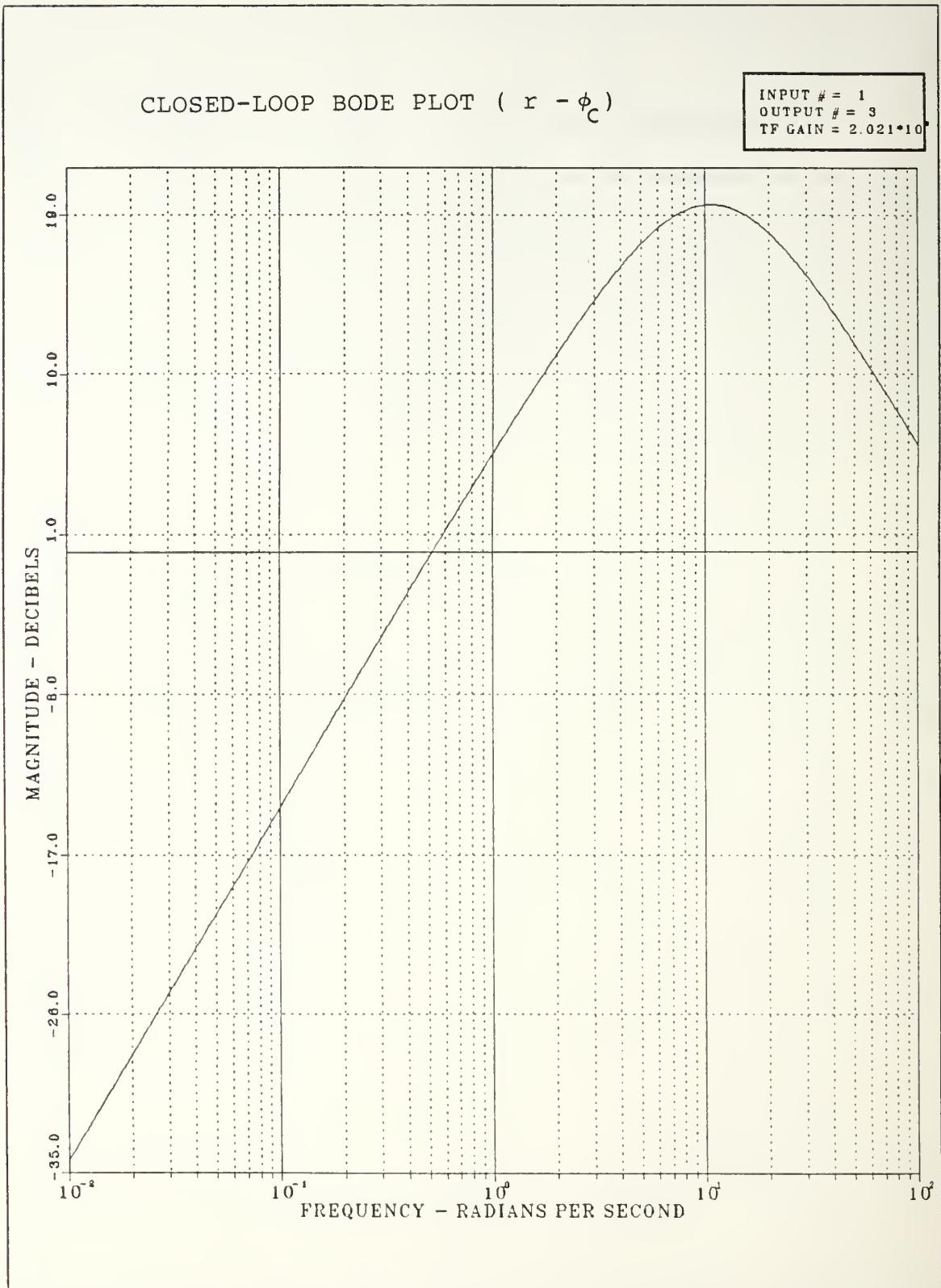


Figure 5.25 Closed-Loop Bode Plot- AlphaTech 1-3.

CLOSED-LOOP BODE PLOT ($\phi - \phi_C$)

INPUT # = 1
OUTPUT # = 4
TF GAIN = $1.129 \cdot 10^4$

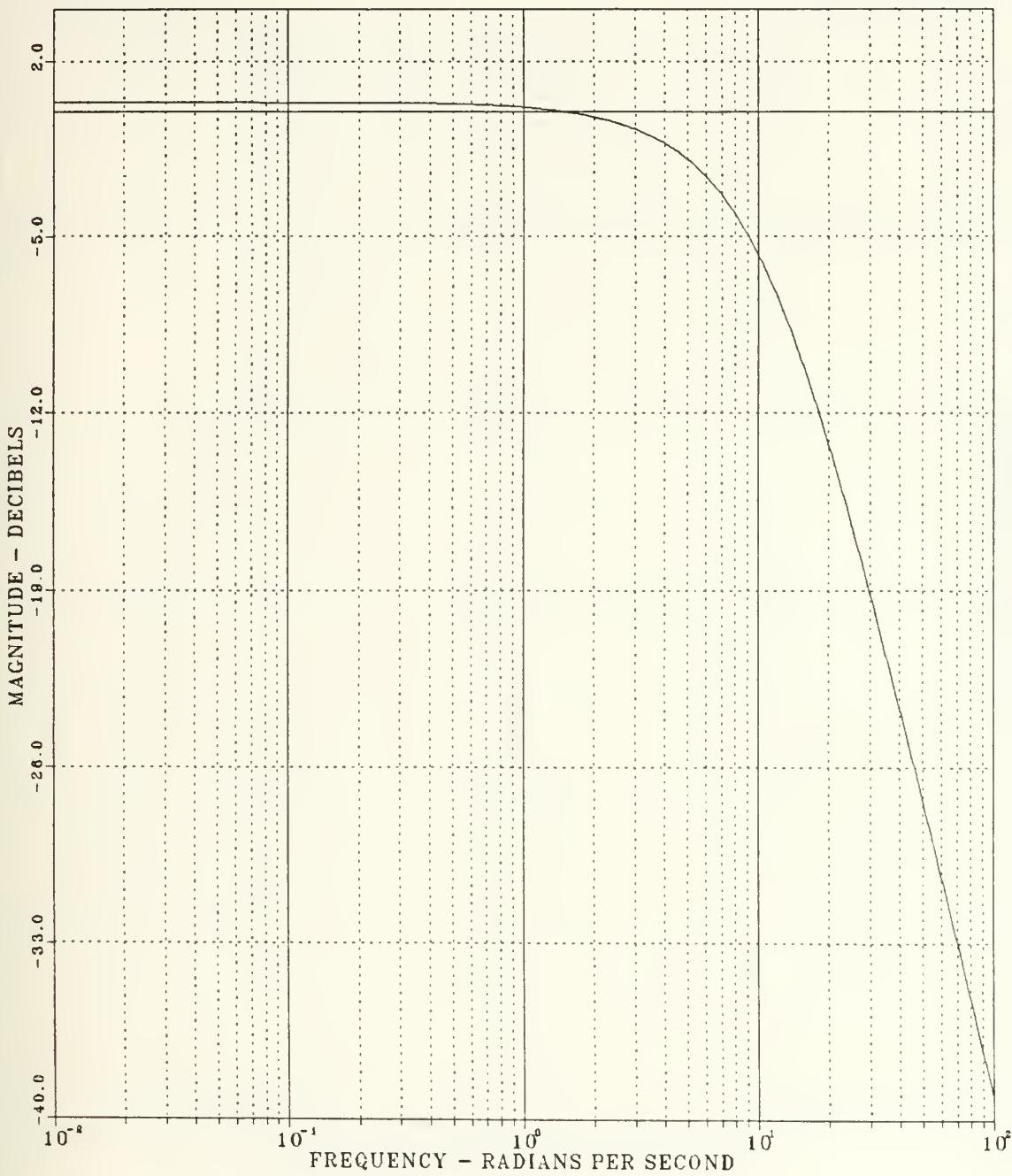


Figure 5.26 Closed-Loop Bode Plot- AlphaTech 1-4.

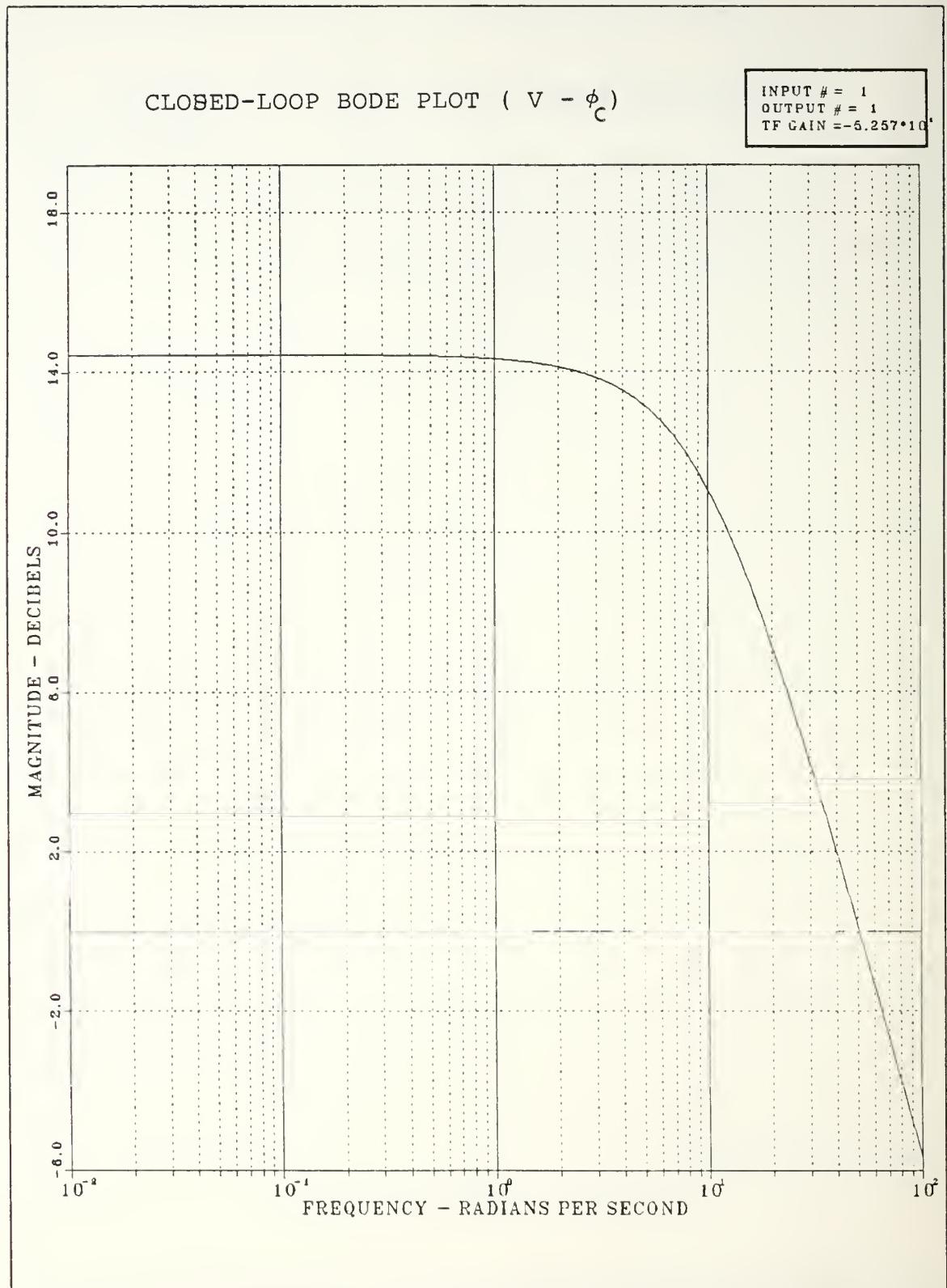


Figure 5.27 Closed-Loop Bode Plot- LQ-A 1-1.

CLOSED-LOOP BODE PLOT ($r - \phi_C$)

INPUT # = 1
OUTPUT # = 3
TF GAIN = $-1.388 \cdot 10^{-3}$

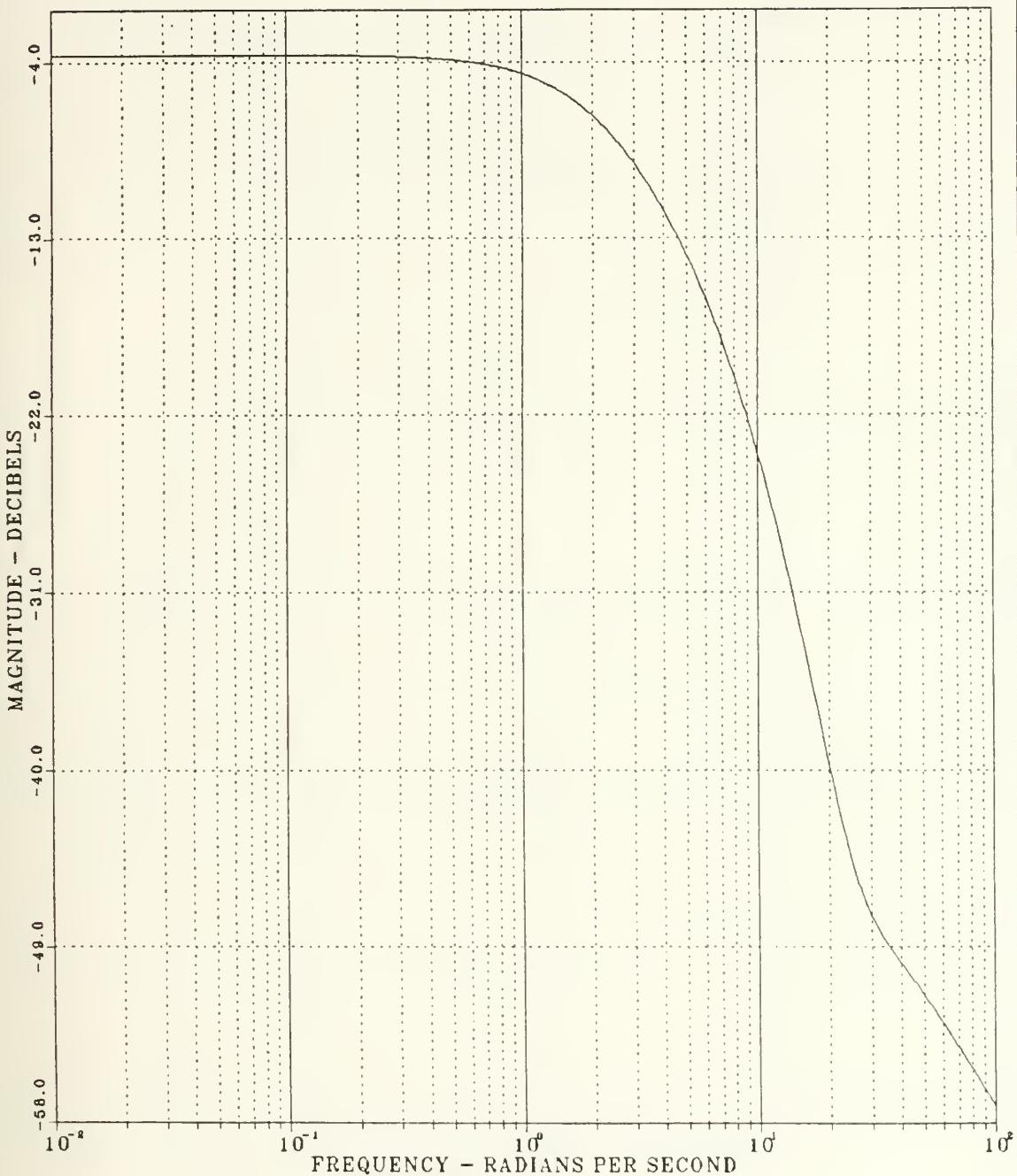


Figure 5.28 Closed-Loop Bode Plot- LQ-A 1-3.

CLOSED-LOOP BODE ($\phi - \phi_c$)

INPUT # = 1
OUTPUT # = 4
TF GAIN = 4.684×10^0

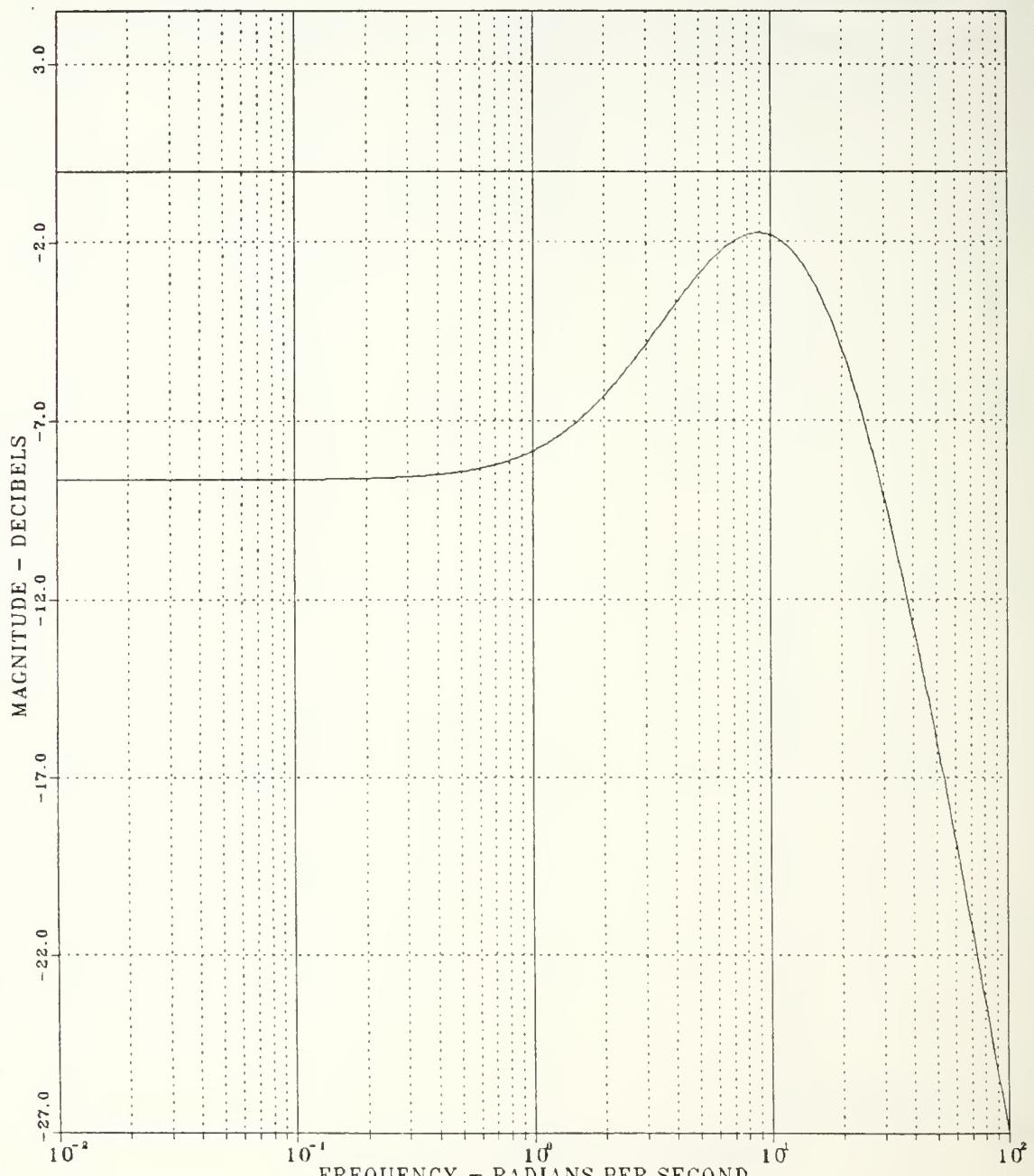


Figure 5.29 Closed-Loop Bode Plot- LQ-A 1-4.

CLOSED-LOOP BODE PLOT ($V - \phi_C$)

INPUT # = 1
OUTPUT # = 1
TF GAIN = $-2.831 \cdot 10^1$

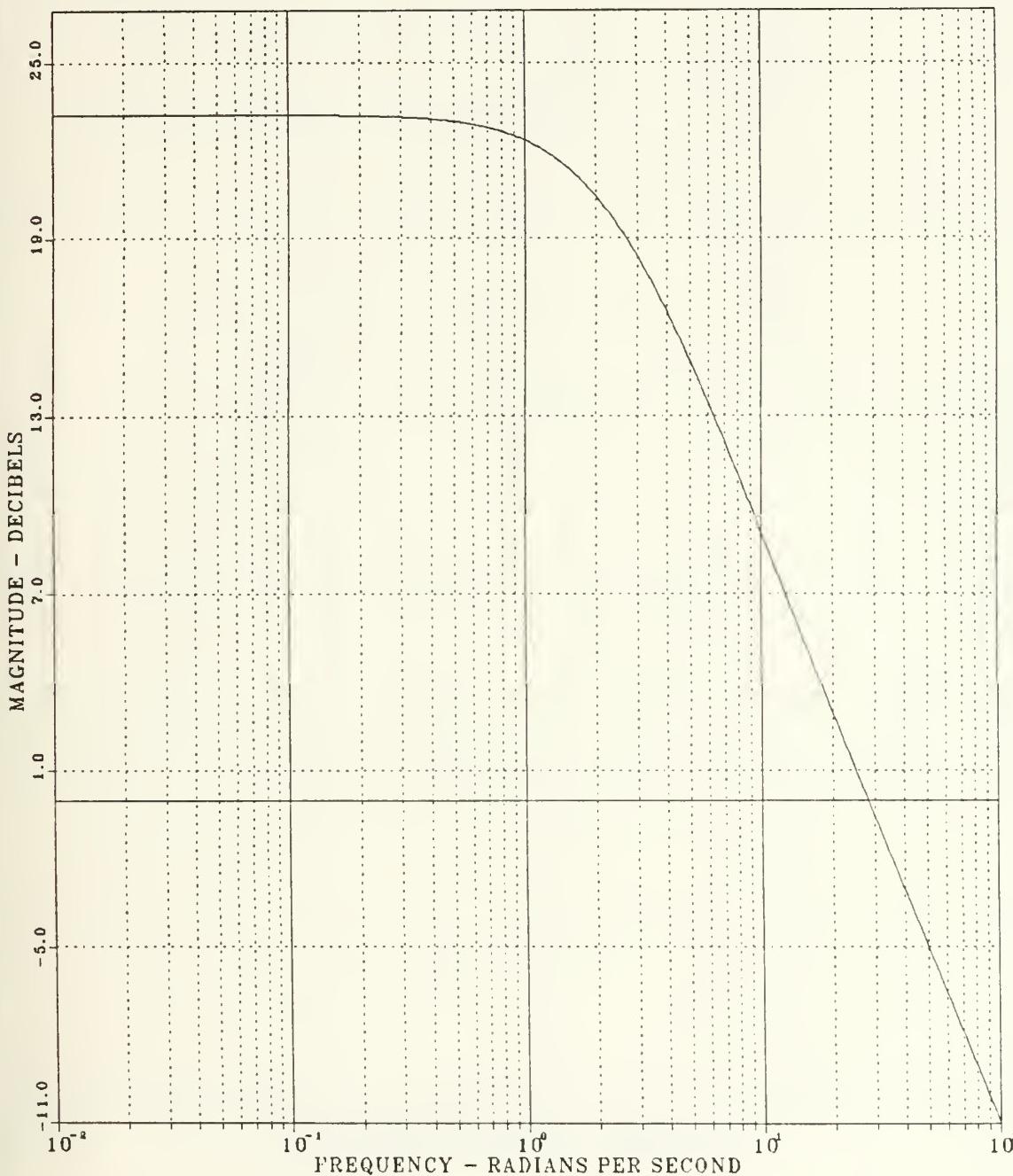


Figure 5.30 Closed-Loop Bode Plot- LQ-B 1-1.

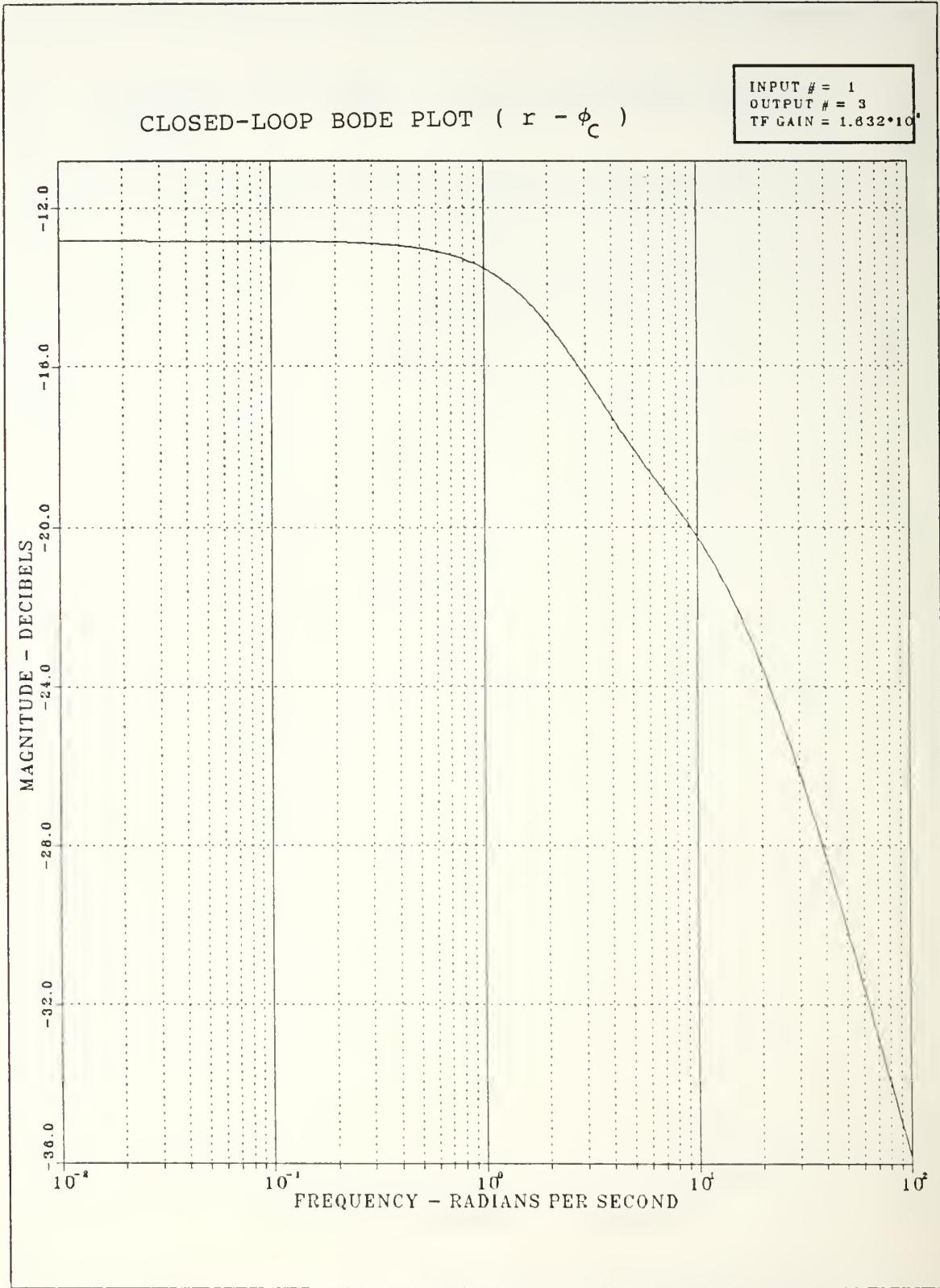


Figure 5.31 Closed-Loop Bode Plot- LQ-B 1-3.

CLOSED-LOOP BODE PLOT ($\phi - \phi_C$)

INPUT # = 1
OUTPUT # = 4
TF GAIN = $2.560 \cdot 10^4$

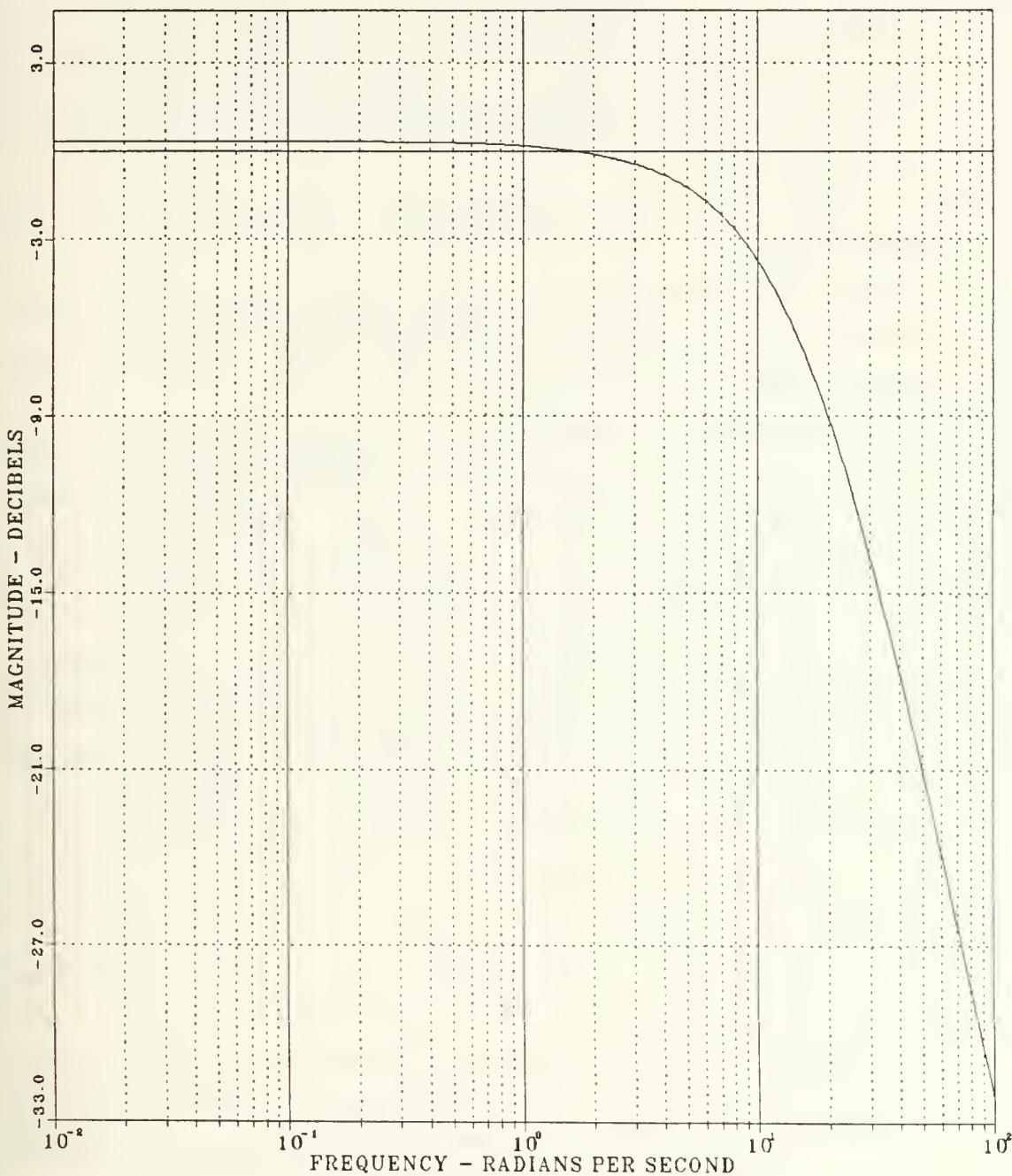


Figure 5.32 Closed-Loop Bode Plot- LQ-B 1-4.

1. It guarantees stability automatically.
2. It has partly overcome the main difficulty in applying optimal control theory to practical problems; i.e. that of selecting suitable performance indices (Q and R matrices). The designer can now choose to compute Q (if he knows somethings about R) or vice versa or compute both Q and R if both of them are unknown.
3. It has built-in robustness to model's error and perturbation.
4. Reduced-order type of problem formulation is possible as the procedure can reassign poles one at a time without affecting others.

On the other hand, the design procedure developed here is not without some shortcomings. Three main problem areas are described below, one of which can be overcome by additional programming efforts while the others are still active research areas pursued by many researchers.

1. Eigenvectors Assignments: It was shown in the design examples presented above that while the overall speed of response of the closed-loop system is determined by its eigenvalues, the 'shape' of the transient response depends on the closed-loop eigenvectors. The problem of eigenvectors assignments was first formulated in [Ref. 16]. Since then many eigenvectors assignment algorithms have been developed [Refs. 17,24] for use in multivariable design. In principle, the eigenvectors assignment routine can be incorporated into the coordinate transformation portion of the OPTPP program (as indicated in Figure 4.1 in Chapter 4). In essence, a new similarity transformation different from M in Equation 4.6 is computed once the eigenvalues and eigenvectors are specified. As most eigenvector algorithms available

- at present are iterative in nature and their inclusion requires major programing efforts, it is recommended for future work.
2. Perturbation or Model's Error Structure: The guaranteed robustness obtained from the LQ formulation given by equations 3.4 and 3.5 ensures that the perturbation or model's error ($L(jw)$) is sufficiently small so that the closed-loop system remains stable. However, the above only applies to simple model's error structure where both $L(jw)$ and R are diagonal matrices. There may be cases when the above equations do not hold and hence the design becomes very conservative. An example of this nature is shown in [Ref. 19]. An ad hoc solution is to use non-diagonal control weighting matrix as mentioned in Chapter 4. Two designs for the helicopter problem are obtained using non-diagonal R . Their results in terms of singular value plots, Bode plots etc are compared with other designs in Appendix A
 3. Reassignment Sequence: As shown in the last section, different reassignment sequences result in different designs. At present there are no known methods for determining the best reassignment sequences.

VI. CONCLUSIONS

It was demonstrated that the general pole assignment problem in multivariable state feedback control system design can be formulated using the Linear Quadratic Control approach. This method of formulation is effective for two main reasons; First, the extra degrees of freedom available in a multivariable system structure is utilized to produce designs that are robust to perturbations in the system and gain matrices. Secondly, the classical difficulty of selecting suitable performance index in optimal control formulation was partly overcome, as designers now have the flexibility of specifying only Q (the state weighting matrix) or R (the control weighting matrix) or both Q and R as the design parameters to be varied. In other words, knowledge of the performance index which ideally should come from physical argument is used to the best of designer's advantage. In addition, the structure of the present formulation is such that eigenvector assignment can be further incorporated into the procedure. The above properties, when combined with the reduced-order formulation capability, have been shown to be very versatile and have important impact on the performance of the resulting design.

The optimal pole placement (OPTPP) program developed here is combined with other well established routines to form a computer aided design and synthesis package. Together with the design procedure and philosophy presented here, it provides the control system designer an excellent and viable tool to solve complex multivariable problems.

The procedure was applied to practical test examples, and numerical results were presented and discussed. Results indicated that all controllers obtained from the formulation

given here were stable and robust. Introducing perturbation in the system matrices leaded only to small errors in the assigned poles. The main shortcoming of the design procedure is the ad hoc nature in which the poles are reassigned. More research is required to develop a systematic way of assigning poles and its eigenvectors thus allowing the designer to optimaly shape the response of the system.

APPENDIX A
NON-DIAGONAL R DESIGNS

The two design examples presented in Chapter 5 were based on diagonal control weighting matrix (i.e. $R = I$). It is now illustrated that design using non-diagonal R will provide yet another degrees of freedom available to the designer. This type of formulation is especially useful when the model's or error structure is known. For multiplicative type of perturbation, the effect of control weighting matrix on the system stability has been explored in [Ref. 12]. In essence, the selection of the R matrix determines the coordinate frame in which the sensitivity assessment is to be made. This can be readily seen from equation 3.1, the general form in which R is non-diagonal is given below;

$$\bar{\sigma} [R^{1/2} L^{-1}(s) R^{-1/2} - I] < \underline{\sigma} [I + G(s)] \quad (\text{eqn A.1})$$

In general, the sensitivity of the system to perturbation in a particular $L(s)$'s direction can be reduced by making $\bar{\sigma} [R^{1/2} L^{-1}(s) R^{-1/2} - I]$ small. This can be done simply by choosing an appropriate R . The main problem with this kind of approach is that $L(s)$ must be known precisely for all frequency for the computation of $\bar{\sigma} [R^{1/2} L^{-1}(s) R^{-1/2} - I]$. In addition, the worst case direction of $L(s)$ must be known otherwise the resulting design may be too conservative.

The helicopter problem presented in Chapter 5 is now analyzed using the non-diagonal control weighting matrix. Two designs are obtained as follows;

Design one (LQ-C):

$$R = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 2.0 \end{bmatrix}$$

Design two (LQ-D):

$$R = \begin{bmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix}$$

Design LQ-C is obtained using an assignment sequence similar to LQ-A with no off-diagonal element in R. Design LQ-D is obtained using placement sequence for design LQ-B but has off-diagonal element in R. Q and F obtained during each move and the final Q_e and F_e are tabulated in Tables VII and VIII. Singular value plots and the open-loop Bode Plots are compared with other designs in Figures A.1 to A.5

Both design has singular value greater than one for all frequency. It is interesting to note that similar reassignment sequence produce similar singular value plots. This can be readily seen by comparing LQ-A with LQ-C and LQ-B with LQ-D in the figures. The effect of using non-diagonal R merely changes the shape of the singular value plots. For the cases presented here, both non-diagonal R designs are slightly less conservative (having lower singular value).

As in the singular value plots, the shape of the open-loop Bode plots (with feedback) are closely related to the reassignment sequence. (compare Bode plots of LQ-A with LQ-C and LQ-B with LQ-D in Figures A.2 to A.5). Results from pole-zero maps also indicate similar trends.

In summary, it is demonstrated that designs using non-diagonal control weighting matrix (including possibly off-diagonal elements) provide yet another means of 'fine

tuning' the design. This capability, such as using the off-diagonal elements directly in design, is unique to the present formulation. Robustness between the upper and lower crossfeed (of $L(s)$) can be controlled by adjusting the relative weighting of the upper and lower (or off-diagonal) elements of the control weighting matrix. It must be emphasized that this kind of fine tuning is only possible for a class of rather well-defined structure of L . In practices, other constraints, such as the energy of the control input, the conditioning of R etc must be considered over the range of frequencies. The key issue of model's error structure and how it can be used in multivariable control system design is currently being investigated by many researchers.

TABLE VII
RESULTS FROM POLE PLACEMENT SEQUENCE (LQ-C)

Move	Q and F obtained during each reassignment			
Q_1	0.00062	0.10604	0.01180	0.09584
	0.10604	18.17288	2.02209	16.42484
	0.01180	2.02209	0.22500	1.82758
	0.09584	16.42485	1.82758	14.84496
F_1	0.00033	0.05708	0.00635	0.05159
	-0.01773	-3.03892	-0.33814	-2.74657
Q_2	4.17258	-3.78394	0.13594	-108.79945
	-3.78394	3.43150	-0.12328	98.66574
	0.13594	-0.12328	0.00443	-3.54455
	-108.79944	98.66574	-3.54455	2836.93335
F_2	0.02701	-0.02449	0.00090	-0.70429
	1.21663	-1.10329	0.03967	-31.72316
Q_3	154.23135	36.25874	26.52098	418.56201
	36.25874	8.52419	6.23491	98.40111
	26.52100	6.23491	4.56044	71.97421
	418.56201	98.40114	71.97421	1135.91846
F_3	2.80842	0.66025	0.48291	7.62180
	-7.58133	-1.78234	-1.30355	-20.57501
Q_4	0.44288	0.09189	-4.22665	0.82784
	0.09189	0.01906	-0.87692	0.17176
	-4.22665	-0.87692	40.33727	-7.90052
	0.82784	0.17176	-7.90052	1.54741
F_4	-0.63449	-0.13165	6.05549	-1.18605
	-0.09950	-0.02064	0.94961	-0.18599
F_e	2.20127	0.56119	6.54565	5.78305
	-6.48193	-5.94519	-0.65241	-55.23073
Q_e	158.84741	32.67273	22.44206	310.68604
	32.67273	30.14761	7.25680	213.66344
	22.44208	7.25680	45.12712	62.35670
	310.68604	213.66348	62.35670	3989.24390

$$u(t) = -Fx(t) + h \phi_C(t), \quad h = \begin{bmatrix} 5.7839 \\ -55.23 \end{bmatrix}$$

TABLE VIII
RESULTS FROM POLE PLACEMENT SEQUENCE (LQ-D)

Move	Q and F obtained during each reassignment			
Q_1	0.00023	0.03940	0.00438	0.03561
	0.03940	6.75195	0.75129	6.10249
	0.00438	0.75129	0.08360	0.67902
	0.03561	6.10249	0.67902	5.51550
F_1	0.00895	1.53374	0.17065	1.38619
	-0.01765	-3.02506	-0.33659	-2.73405
Q_2	0.24621	0.11866	14.98355	9.73271
	0.11866	0.05719	7.22105	4.69050
	14.98355	7.22105	911.83813	592.29370
	9.73272	4.69050	592.29370	384.73022
F_2	0.56910	0.27424	34.63293	22.49620
	-0.32115	-0.15476	-19.54404	-12.69504
Q_3	0.17288	-0.43406	7.14036	-7.87012
	-0.43406	1.08987	-17.92834	19.76067
	7.14036	-17.92834	294.92139	-325.06299
	-7.87012	19.76067	-325.06299	358.28516
F_3	0.05761	-0.14465	2.37951	-2.62273
	0.34047	-0.85487	14.06264	-15.49990
F_e	0.63566	1.66333	37.18307	21.25964
	0.00167	-4.03469	-5.81798	-30.92899
Q_e	0.41932	-0.27600	22.12828	1.89820
	-0.27600	7.89901	-9.95601	30.55365
	22.12828	-9.95601	1206.84302	267.90967
	1.89821	30.55365	267.90967	748.53076

$$u(t) = -F_x(t) + h \phi_C(t), \quad h = \begin{bmatrix} 21.159 \\ -30.9289 \end{bmatrix}$$

SINGULAR VALUE PLOT

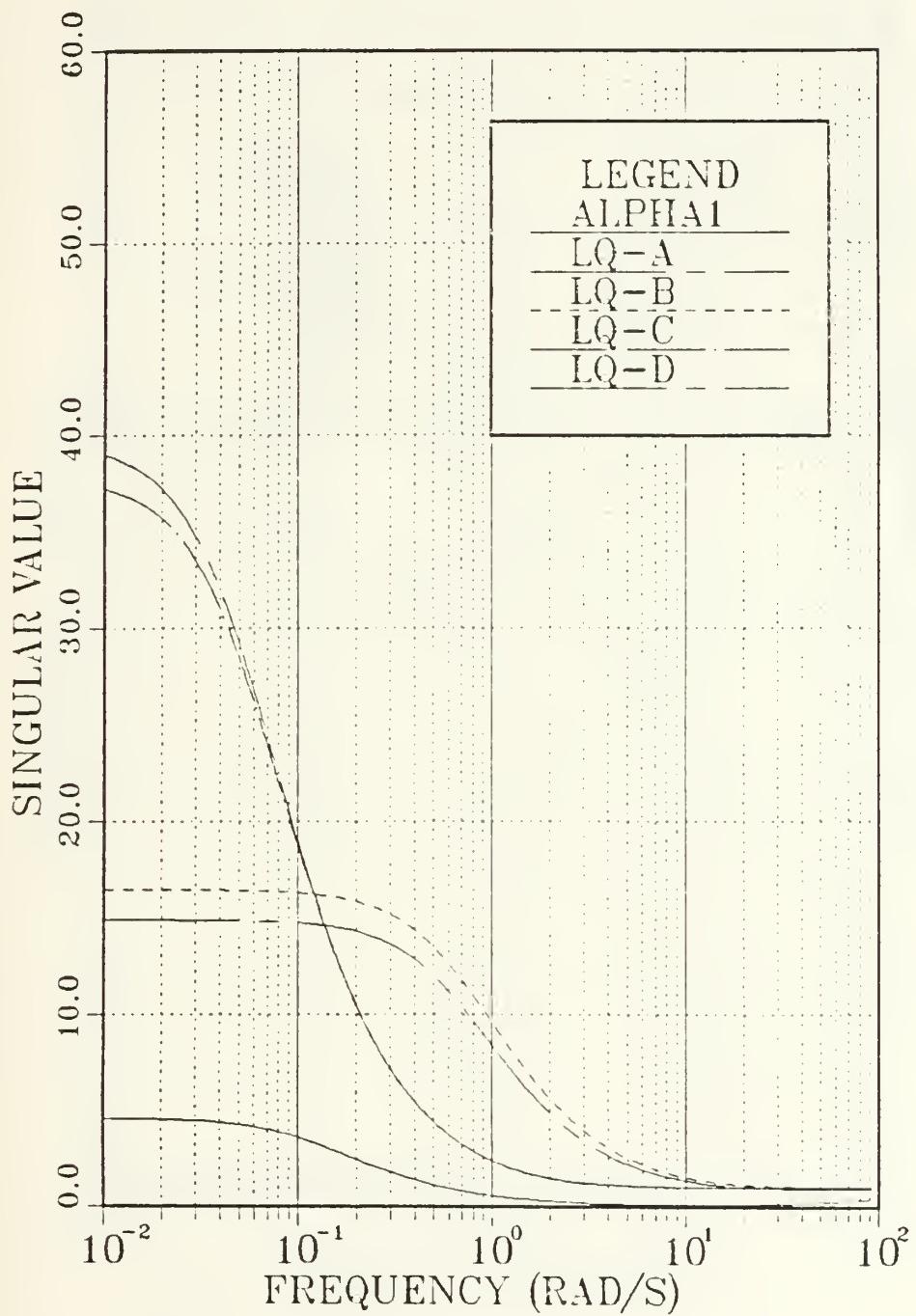


Figure A.1 Singular Value Plots - Comparison.

OPEN LOOP GAIN 1 - 1

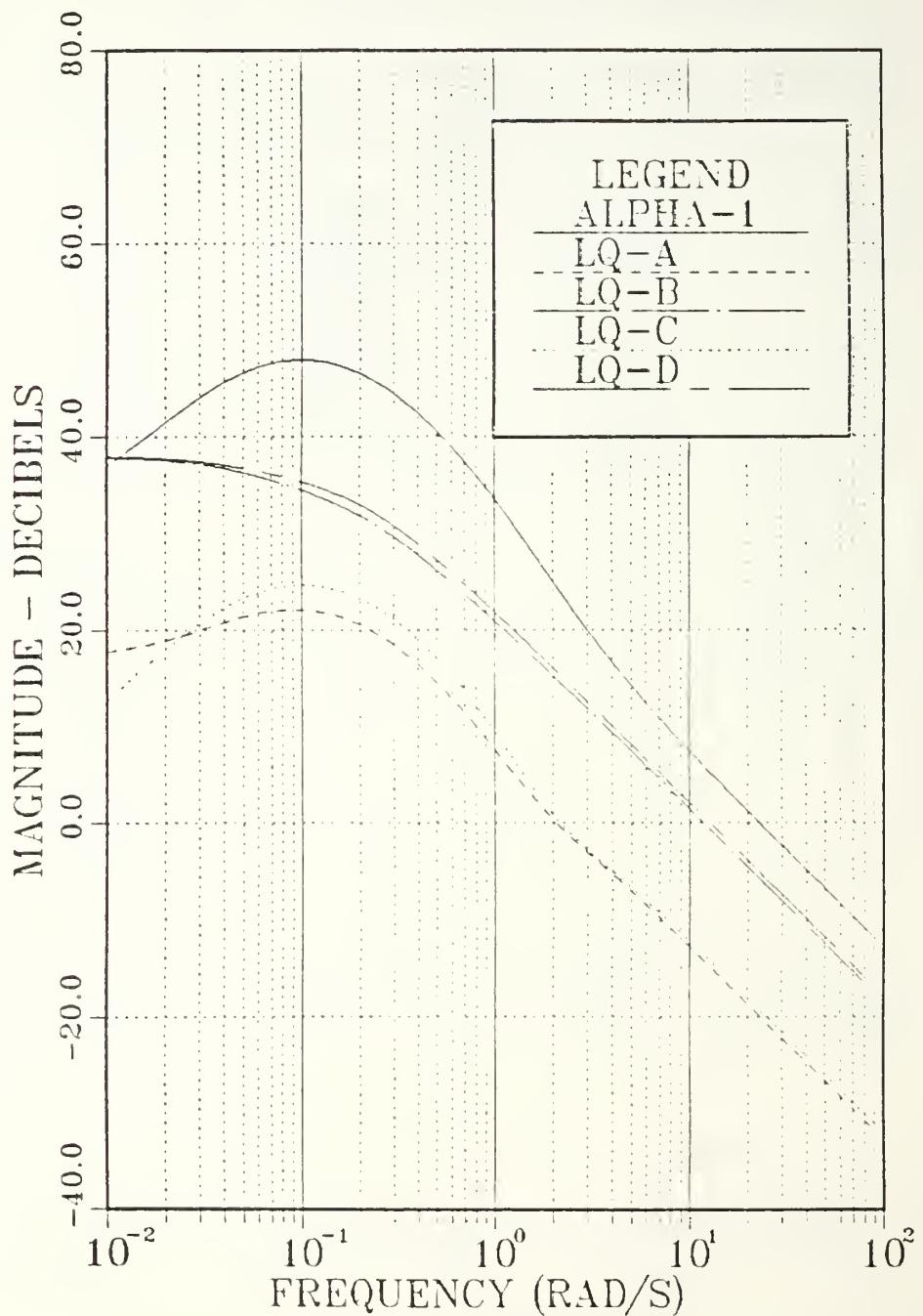


Figure A.2 Bode Plots Comparison- Input 1-1.

OPEN LOOP GAIN 1-2

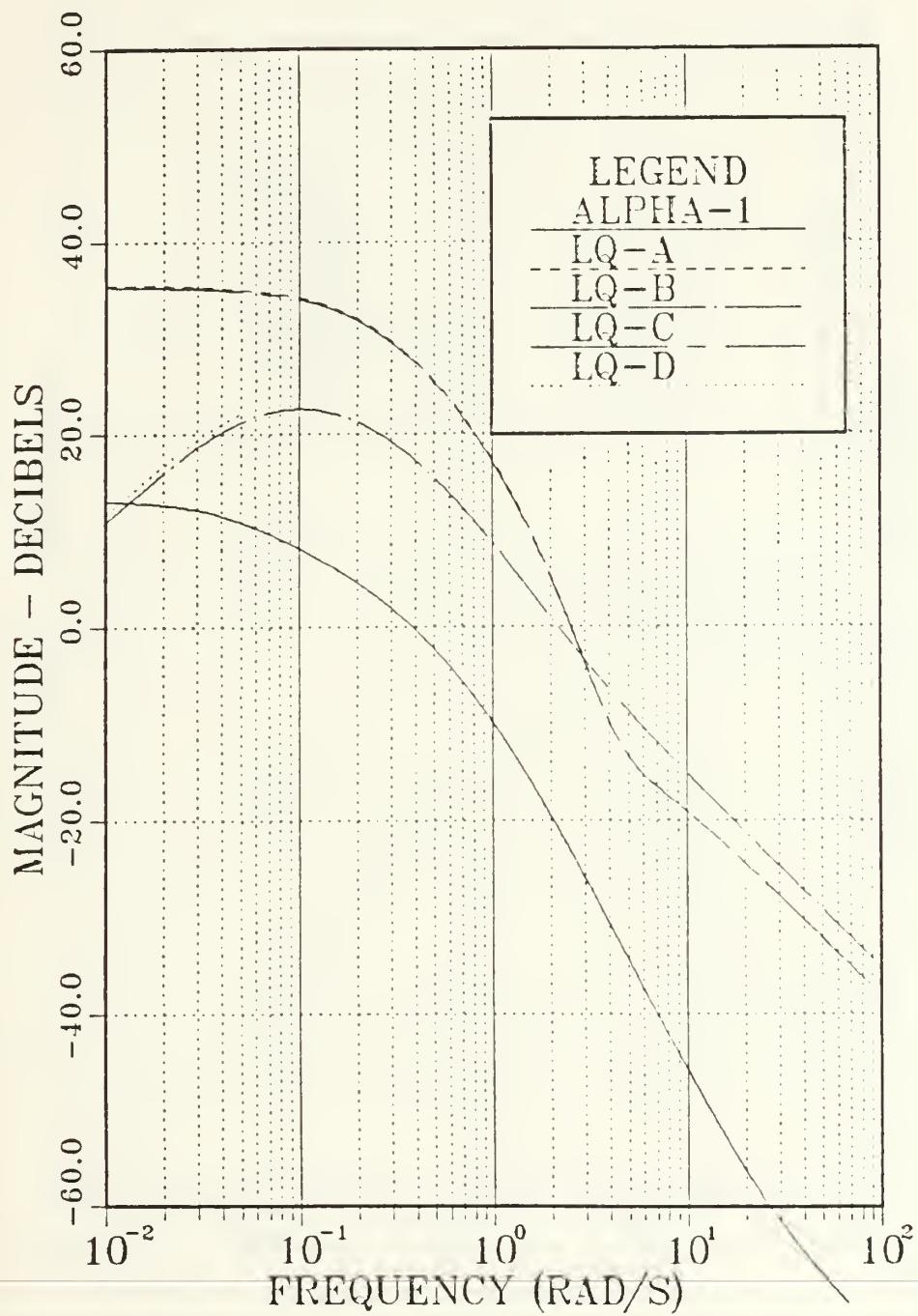


Figure A.3 Bode Plots Comparison - Input 1-2.

OPEN LOOP GAIN 2-1

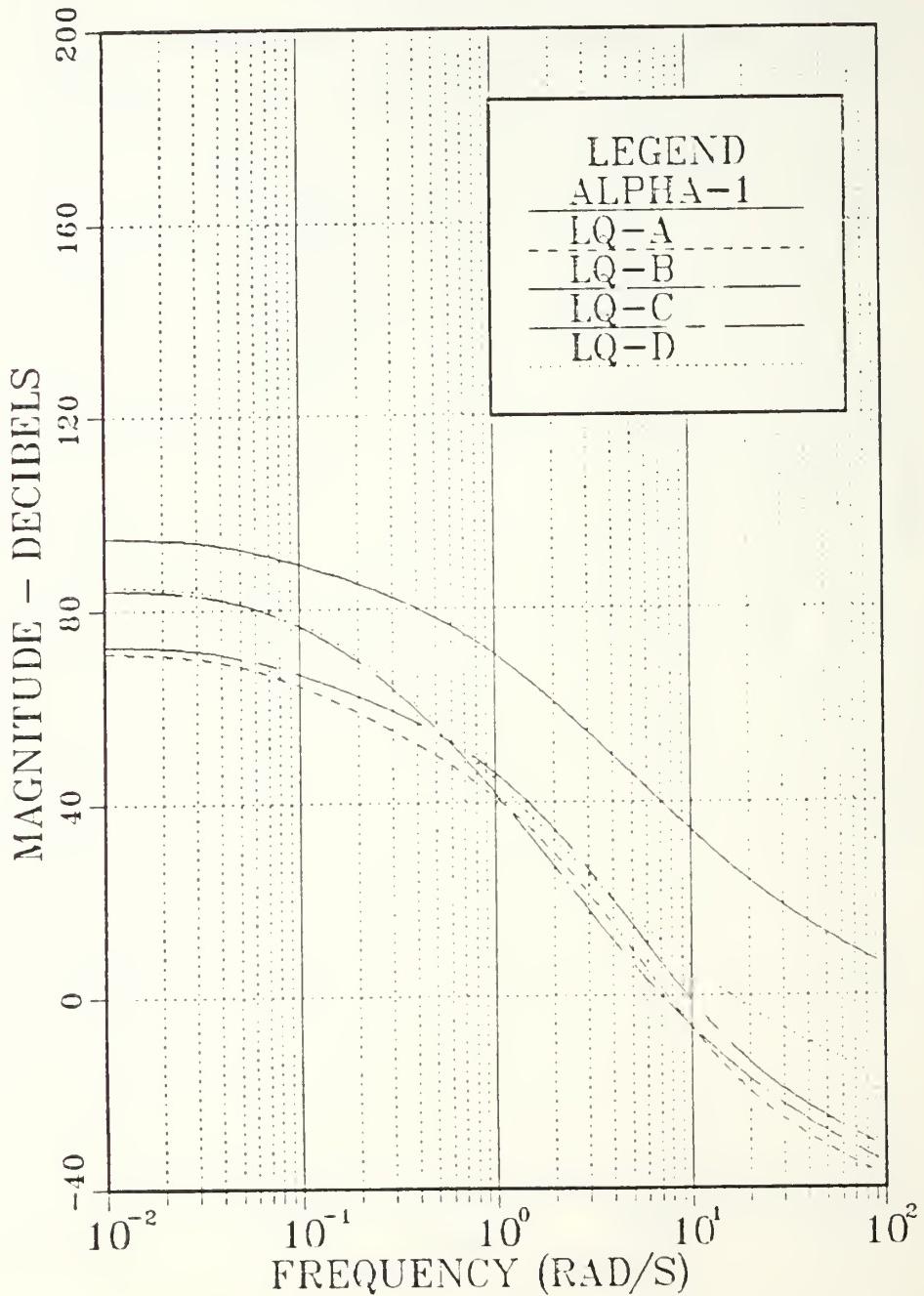


Figure A.4 Bode Plots Comparison - Input 2-1.

OPEN LOOP GAIN 2-2

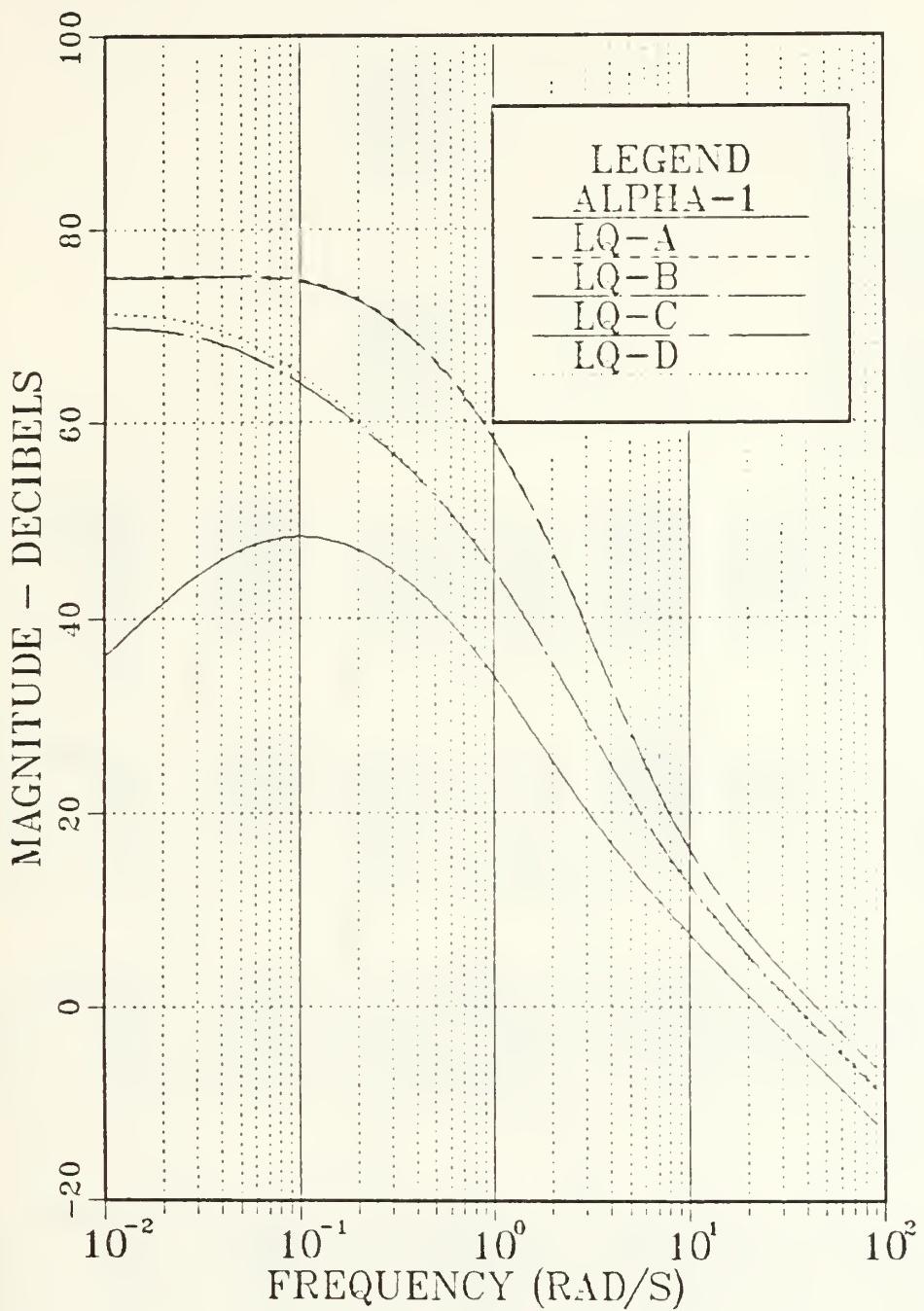


Figure A.5 Bode Plots Comparison - Input 2-2.

APPENDIX B : EXAMPLE DESIGN RUN OUTPUTS

*** THIS EXAMPLE DEMONSTRATES THE POLE REASSIGNMENT WHERE THE OPEN LOOP POLE -2.146 IS MOVED TO ITS NEW LOCATION AT -25.21 ***
*** THE DESIGN VARIABLE THAT THE UNSTABLE POLES ***
*** AUTOMATICALLY MOVED TO ITS MIRROR IMAGE DUE TO LQ ***
*** C.2005 FORMULATION. ***

KONTROL

0

*** THE A PLANT MATRIX ***

-2.47000	1.42000	-0.15000	31.99001
0.04000	-0.70000	-0.07000	C.00000
0.04000	-0.05000	-0.05000	C.00000
0.00000	1.00000	0.11000	C.00000

*** THE B CONTROL INPUT MATRIX ***

C.12000	C.95000		
0.04000	-6.37000		
0.34000	C.02000		
C.00000	C.00000		

*** THE STARTING Q WEIGHTING MATRIX ***

0.00000	C.00000	0.00000	C.00000
0.00000	C.00000	0.00000	C.00000
0.00000	C.00000	0.00000	C.00000
0.00000	C.00000	0.00000	C.00000

*** THE DESIGN VARIABLE (Q) X ***

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	1

```

*** THE R WEIGHTING MATRIX ***
1.00000 0.00000
0.00000 1.00000

*** THE STARTING/FINAL F MATRIX ***
-14.78000 -2.16000 77.97000 -55.30000
-0.00567 -2.55600 0.39400 -15.05000

*** THE ORDERED COMPLEX EIGENVALUES (INPUT)
-25.21001 C.00000

*** ICOMP ***
0

*** AUX TRANSFORMATION L ***
0.00000 0.00000 C.00000
0.00000 0.00000 0.00000 C.00000
0.00000 0.00000 0.00000 C.00000
0.00000 0.00000 0.00000 C.00000
EIGENVECTOR OF A ***
0.14468 C.98509 0.99767 0.99777
0.10820 -0.00101 -0.03687 -0.00796
-0.98384 C.15380 -0.04176 -C.01945
C.00049 0.07703 0.03949 C.00475

*** TRANSFORMED A MATRIX ***
-0.5030 C.00000 0.00000 0.00000
0.00000 Q.20653 0.00000 C.00000
0.00000 -C.00001 -1.04987 -C.00001
0.00008 C.00001 0.00002 -2.12635
*** TRANSFORMED Q MATRIX ***
0.00000 C.00000 0.00000 0.00000
0.00000 0.00000 0.00000 C.00000
0.00000 0.00000 0.00000 0.00000
0.00000 C.00000 0.00000 C.00000
*** TRANSFORMED E MATRIX ***
-0.15647 -2C.48637 -91.42015
0.85855 -1.80459 19.07515
-1.09721 -9.72533
*** SQUARE ROOT INVERSE CF R ***
1.00000 0.00000 1.00000

```

AAAAA DDDDD SSSSS
A A D D
A A D D
A A D D
A A D D
A A D D
A A D D

F O R T R A N P R O G R A M

F O R

A U T O M A T E D D E S I G N S Y N T H E S I S
V E R S I O N 1.00

CONTROL PARAMETERS
ISTRAT = 5 IOPT = 3 IONED = 3 IPRINT = 1000
IGRAD = 0 NDV = 1 NCCN = 2

OPTIMIZATION RESULTS

OBJECTIVE FUNCTION VALUE 0.56055E-13

DESIGN VARIABLES

VAR TABLE	LOWER BOUND	VALUE	UPPER BOUND
1	0.0000E+00	0.6858E-01	0.10000E+05

DESIGN CONSTRAINTS

1) -C.5000E-01 -0.9000E+00

FUNCTION EVALUATIONS = 41

```
*** THE OPTIMAL Q MATRIX
 0.00000  0.00000  0.00000  0.00000
 0.00000  0.00000  0.00000  0.00000
 0.00000  0.00000  0.00000  0.00000
 0.00000  0.00000  0.00000  0.06886
```

END OF POLE PLACEMENT ROUTINE

THE Q REQUIRED FOR THIS ASSIGNMENT IS

0.10027	C.96194	0.11959	-1.50849
0.9194	9.22847	1.14725	-14.47182
0.11959	1.14725	0.14262	-1.79908
-1.50849	-14.47182	-1.79908	22.69429

EIGENSYSTEM OF OPTIMAL REGULATOR.....

```
C-LOOP OPTIMAL REG. E-VALUES...DET(SI-F+G#C)..
-2.52094E+01:-1.04978E+00:-2.00591E-01:-5.02113E-02:
CONTROL EIGENVECTOR MATRIX...C*M...
-3.326553E-C2 -1.147587E-08 -6.642234E-04 -4.134316E-06
. 2.902164E+00 -1.370907E-06 -2.350589E-03 -5.652895E-06
```

*** THE STARTING/FINAL F MATRIX ***

0.00387	0.03151	0.00399	-0.06416
-0.24260	-2.83522	-0.34541	2.11642

*** THE AUGMENTED A-B*C MATRIX ***

-2.03999	4.10968	0.17766	29.03711
-2.02074	-24.43202	-2.96120	26.08698
0.04354	-0.00401	-0.04445	-0.04051
0.00000	1.00000	0.11000	0.00000

END OF CPTPP ANALYSIS

```

C***** INPUT DATA FILE *****
C
C
C

```

00000C	0	.C1	.01					
100C	i.0	i.0	i.0					
1.0	1.0	1.0	1.0					
0	1	2	5	0				
04040402040402040402040201	0	3	1000	0				
-2.27	1.420	1.420	-0.15		31.99			
0.01	-0.7	-0.07	0.05		0.0			
0.04	-0.05	-0.05	0.11		0.0			
0.0	0.1	0.1	0.					
0.12	0.95	0.95	0.					
0.04	-0.87	-0.87						
0.34	0.20	0.20						
0.0	i.0	i.0						
0.1.	C.	C.	0.	0.	0.			
0.0.	1.	1.	1.	0.	0.			
0.	C.	C.	1.	1.	1.			
						-55.3		
-0.4.78	2.16	77.97	0.394					
-0.00567	-2.556	0.00	0.00					
0.00	0.00	0.00	0.00					
0.00	0.00	0.00	0.00					
1.000	C.00	1.00	0.					
0.000	0.000	0.000	0.					
0.000	0.000	0.000	1.000					
0.000	0.000	0.000	10000.					
-25.21								

```

***** THIS EXAMPLE DEMONSTRATES THE POLE REASSIGNMENT WHERE THE OPEN
***** LOOP COMPLEX CONJUGATE PAIR OF POLE AT (-0.39354, 1.426)
***** IS MOVED TO ITS NEW LOCATION AT (-0.7, 1.42). Q(1,1) AND
***** Q(2,2) ARE THE DESIGN VARIABLE. THERE ARE NO UNSTABLE POLES
***** IN THIS CASE SO THE OTHER POLES/EIGENVECTORS REMAIN UNCHANGE.
*****
```

```

KONTROL      0
*** THE A PLANT MATRIX ***
-0.18000    -C.03000    9.57000   -31.87000
-0.20000    -C.55000    109.42999   2.78000
-0.01000    -C.01770   -0.10000   C.00000
0.00000     C.00000    1.00000   C.00000
```

```

*** THE B CONTROL INPUT MATRIX ***
-0.35600    0.52000
0.00000    -1.00000
0.33600    -0.02100
0.00000     C.00000
```

```

*** THE STARTING Q WEIGHTING MATRIX ***

```

0.00000	0.00000	0.00000	0.00000
0.00000	C.00000	0.00000	C.00000
0.00000	0.00000	0.00000	C.00000
0.00000	0.00000	0.00000	C.00000

```

*** THE DESIGN VARIABLE (Q) X ***

```

1	0	0	0
0	2	0	0
0	0	0	0

AAAAAA DDDDDD SSSSSS
 A A D D S
 AAAAAA D D SSSSSS
 A A D D S
 A A D D S
 A A D D S
 A A D D S

F O R T R A N P R C G R A M
 F O R
 A U T O M A T E D D E S I G N S Y N T H E S I S
 V E R S I O N 1.00

CONTROL PARAMETERS
 ISTRAT = 5 IOPT = 3 IONED = 3 IPRINT = 1000
 IGRAD = 0 NDV = 2 NCEN = 3

OPTIMIZATION RESULTS-----

OBJECTIVE FUNCTION VALUE 0.28365E-10

DESIGN VARIABLES

VARIABLE	LOWER BOUND	VALUE	UPPER BOUND
1	0.0000E+00	0.23088E-02	0.10000E+04
2	0.0000E+00	0.22134E-02	0.10000E+04

DESIGN CONSTRAINTS

1) -C.4999E-01 -C.9000E+00 0.9545E-04
 FUNCTION EVALUATIONS = 48

```

***** THE OPTIMAL Q MATRIX
      0.000231    C.000000    0.000000    C.000000
      0.000000    C.00221     0.000000    C.000000
      0.000000    C.000000    0.000000    C.000000
      0.000000    C.000000    0.000000    C.000000
END OF POLE PLACEMENT ROUTINE

THE Q REQUIRED FOR THIS ASSIGNMENT IS
      0.00060     0.00106    -0.00908    C.00301
      0.00106     0.00189    -0.01970    C.00579
      -0.00908    -0.01970   12.94063   -0.60992
      -0.00301    -0.00579   -1.60992   0.20628

EIGENSYSTEM OF OPTIMAL REGULATOR..... .
C-LQCP OPTIMAL REG. E-VALUES... .DET(SI-F+G*C)...
-6.99887E-01, 1.41996E+00:-1.80579E-01:-1.37682E-01:

CONTROL EIGENVECTOR MATRIX..... .C*M..
4.996814E-03 -2.528824E-02 -6.565824E-08 -5.181365E-03
2.956046E-C4 -4.560151E-04 -3.259629E-09 -1.896438E-04

***** THE STARTING/FINAL F MATRIX ****
      -0.01311    -0.02106    2.67146    2.79805
      -0.00052    -0.00084    0.06034    C.09177

***** THE AUGMENTED A-B*C MATRIX ****
      -0.18440    -0.03706    10.48966   -3.0.92160
      -0.20052    -0.55084    109.49023   -2.87177
      -0.00566    -0.01073   -0.98285   -C.92528
      0.00000    C.00000    1.00000   C.00000
END OF CPTPP ANALYSIS

```

***** INPUT DATA FILE *****

APPENDIX C : COMPUTER PROGRAM LISTINGS

*** CPTPP --- OPTIMAL POLE PLACEMENT PROGRAM***

THIS PROGRAM IS DESIGNED TO COMPUTE THE STATE WEIGHTING MATRIX OF A MINIMUM CONTROL PROBLEM GIVEN THE DESIRED CLOSED-LOOP POLES LOCATIONS. THE DESIGNER HAS THE CHOICE OF ASSIGNING ONE OR MORE POLES DURING ONE RUN. ON COMPLETION OF THE OPTYSIS ROUTINE IS CALLED TO COMPUTE PLOTS ARE THEN COMPUTED TO DETERMINE THE SYSTEM ROBUSTNESS.

THE PROGRAM CAN BE MODIFIED TO COMPUTE THE CONTROL WEIGHTING MATRIX FOR BOTH THE STATE AND CONTROL WEIGHTING MATRIX IF BOTH OF THEM ARE UNKNOWN.

THE EIGENVECTOR TYPE OF ASSIGNMENT CAN BE INCORPORATED IF REQUIRED

BY CHOK YAP KEH, REPUBLIC OF SINGAPORE
VERSIGN 1.0 SEPT 1985

*** THIS FIRST BLOCK SETS DIMENSIONS AND DECLARIES REAL AND COMPLEX FUNCTIONS. THE FOLLOWING IS A LIST OF PROGRAM SYMBOLS:

REAL VARIABLES BY NAME AND BRIEF DESCRIPTION

THE SYSTEM UNDER CONSIDERATION IS GIVEN AS:
 $\dot{x} = Ax + Bu$; $y = Cx$
 OUTPUT FEEDBACK FORM $U = -Fx + R$ OR $U = -Fy + R$

- A = THE PLANT MATRIX ($N \times N$)
- B = THE CONTROL MATRIX ($M \times N$)
- C = THE PLANT OBSERVATION MATRIX ($M \times N$)
- F = THE FEEDBACK MATRIX ($M \times M$)
- R = THE FEEDBACK MATRIX ($M \times N$)

THE STATE WEIGHTING MATRIX ($N \times N$)

THE CONTROL WEIGHTING MATRIX ($M \times M$)

THE DESIRED CLOSED-LOOP POLES LOCATIONS

THE REAL PART OF THE DESIRED POLE LOCATION

THE COMPLEX PART OF THE DESIRED POLE LOCATION

FC = F^*C
 $BFC = B^*F^*C$
 $AMBFC = A - B^*F^*C$

CONOC110
 CONOC120
 CONOC130
 CONOC140
 CONOC150
 CONOC160
 CONOC170

CONOC200
 CONOC210
 CONOC220
 CONOC230
 CONOC240
 CONOC250
 CONOC260
 CONOC270
 CONOC280
 CONOC290
 CONOC300
 CONOC310
 CONOC320
 CONOC330
 CONOC340
 CONOC350
 CONOC360
 CONOC370

CONOC280
 CONOC290
 CONOC400
 CONOC410
 CONOC420

REIG - REAL COMPUTED EIGENVALUE OF THE SYSTEM

CONOC430

THE FOLLOWINGS ARE USED BY THE SINGULAR VALUE ANALYSIS MODE *

OMEGA - FREQUENCY
 SV - SINGULAR VALUE OF $(I + F * INV(F * G))$ PLANT TRANSFER FUNCTION (G)
 SVMI - SINGULAR VALUE OF $(I + INV(F * G))$
 SSVI - SINGULAR VALUE OF $(I + INV(F * G))$
 SSVC - ONLY OF INTEREST IF NO. OF INPUTS = NO. OF INPUTS.
 SVMC - FIRST SINGULAR VALUE OF $(I + F * G)$
 - SECOND SINGULAR VALUE OF $(I + F * G)$
 - MAXIMUM SINGULAR VALUE OF $(I + F * G)$
 SIGNM1 - SQR(FIRST VAL * SECOND VAL * SING VAL) *
 SIGNM2 - SQRT(FIRST ADDITIVE OUTPUT SINGULAR VALUE
 - MIN A ADDITIVE OUTPUT SINGULAR VALUE
 - MAX A ADDITIVE OUTPUT SINGULAR VALUE
 - MIN M MULTIPLICATIVE INPUT SINGULAR VALUE
 - MAX M MULTIPLICATIVE INPUT SINGULAR VALUE
 - MIN M MULTIPLICATIVE OUTPUT SINGULAR VALUE
 - MAX M MULTIPLICATIVE OUTPUT SINGULAR VALUE

```

IMPLICIT REAL*4 (A-H,O-Z)
REAL#4 (10,10),B(10,10),C(10,10),REALMU(50),
FC(10,10),BF(10,10),AM,BFC(10,10),WK(4000),MEIG(10),
IEIG(50),SV(10,10),WM(10,10),WAL(100),MINEIG(500),
SIGNM(500),SIGPRO(500),
SIGNM(500),SVMM(500),SIGMIX(500),
SIGNM(500),SVMM(500),SVMM(500),LP21(500),
SVADM(500),SVADXO(500),LP22(500),
SVADM(500),SVADXO(500),SVAC(10),SVMD(10),TEMPR(10,10),
SVMM(500),SVAC(10),SVMD(10),RWSPI(10,10),RCOND,ZI(10),
QW(10,10),RW(10,10),RWSPI(10,10),RCOND,ZI(10)

```

***** THE COMPLEX PARAMETERS USED IN THE PROGRAM *****
 AX - COMPLEX A
 BX - COMPLEX B
 CX - COMPLEX C
 FX - COMPLEX F
 QW - COMPLEX OF QW
 RW - COMPLEX OF RW
 RSQX - COMPLEX OF R#*(-1/2)

- EIGENVECTOR OF A (MIN THE THESIS)
- THE FOLLOWINGS ARE USED IN THE COORDINATE TRANSFORMATION BLOCK
- CCNOC 790

ب ب ب ب ب ب ب ب ب ب

```

C21 - INVERSE OF Z TRANSPOSE OF TRANSFORMATION MATRIX FOR COMPLEX EIGENVALUE
C2T1 - INVERSE OF TRANSFORMATION MATRIX FOR COMPLEX EIGENVALUE
LX - AUX TRANSFORM OF LX
LX1 - INVERSE OF LX
ZL1 - INVERSE OF (Z*LX)
-- THE FOLLOWINGS ARE USED TO COMPUTE THE OBJECTIVE FUNCTION
-- GPS - COMPLEX MATRIX COMPUTE BY PLANT1 G(S)
-- WHERE G(S)=C*(S1-A)**(-1) * B
-- SAME AS ABOVE BUT G(-1/2)
-- GPSRG - G(S)*R*(-1/2)
-- GPSRG - C*GPSRSQ
-- GGR - GM*S*QGR
-- R**(-1/2) * GQGR
-- XIPGR - I + RGQGR
-- DETERM - AN ARRAY CONSISTING OF THE DET OF XIPQI
-- TO BE USED IN THE OBJECTIVE FUNCTION OF THE OPTIMIZER

-- THE FOLLOWING ARE USED BY THE SINGULAR VALUE ANALYSIS
-- EIG - EIGENVALUE OF A- B*F*C
-- MU - INPUT DESIRED POLE LOCATION
-- OMU - ORDERED VALUE OF MU
-- OEG - ORDERED EIGENVECTOR
-- CZ - IDENTITY MATRIX
-- XX1 - IDENTITY MATRIX
-- U1 - INPUT ADDITIVE SINGULAR VECTOR ASSOCIATED WITH SV
-- V1 - INPUT ADDITIVE SINGULAR VECTOR ASSOCIATED WITH SV
-- UC - OUTPUT ADDITIVE SINGULAR VECTOR (SVAO)
-- UMI - INPUT MULTIPLICATIVE SINGULAR VECTOR (SVAO)
-- VMI - INPUT MULTIPLICATIVE SINGULAR VECTOR
-- UMO - OUTPUT MULTIPLICATIVE SINGULAR VECTOR
-- VMO - OUTPUT MULTIPLICATIVE SINGULAR VECTOR
-- FXPLT - F * PLANT TRANSFER FUNCTION (G)
-- XIPFXP - I + F*G
-- PLTFX - PLANT TRANSFER FUNCTION (G)
-- XIPPFX - I + G*F
-- PLTFFX - I + INV(F*G)
-- FXPLTI - INV(G*F)
-- XIPFLI - I + INV(F*G)
-- XIPLF1 - I + INV(F*F)

```


MODIFIED BY PROGRAM AT EACH DECADE SHIFT
 CARD 3 EIGHTING WT3; FORMATTED 3F10.0
 EIGHTING FACTORS TO BE USED IN THE MULTIPLE POLES
 PLACEMENT MODES (NOT USED IN THE PRESENT FORMULATION)
 CARD 4 SUMINI, RJSNIDG; FORMAT(3F10.0 15)
 ONLY NIDG IS USED IN THIS PROGRAM
 SUMINI - DESIRED SINGULAR VALUE LEVEL FOR THE S.V. OF
 SUMINI (1+F) FOUND USING UNIVERSAL GAIN AND PHASE CHART
 TO GIVE A DESIRED PHASE AND GAIN MARGIN
 SUMINO - OUTPUT VERSION OF SUMINI
 RJ - FACTOR TO SET PERCENT POLE PLACEMENT ERROR THE
 DESIGNER WISHES TO PUT INTO THE OPTIMIZER
 ----- NIDG - PARAMETER THAT SETS TYPE CONSTRAINT CONSIDERED
 SEE ADS MANUAL
 CARD 5 IGRAD,NDV,NCON,ISTRAT,IOPT,IONED,IPRINT,INFO
 IGRAD - PARAMETER CONCERNING GRADIENT COMP SEE ADS
 NDV - NUMBER OF DESIGN VARIABLES
 NCON - NUMBER OF CONSTRAINTS SEE ADS
 ISTRAT - OPTIMIZER STRATEGY SEE ADS MANUAL
 IOPT - OPTIMIZATION METHOD SEE ADS
 ICNED - ONE DIMENSIONAL SEARCH TECHNIQUE SEE ADS
 IPRINT - ADS PRINT CONTROL PARAMETER SEE ADS
 INFO - OPTIMIZER CONTROL PARAMETER SEE ADS
 CARD 6 NROWA,NCOLB,NCOLB,NROWF,NCOLC,NROWC,
 NCOLQ,NROWF,NCOLR,NMU,
 FCNRA(1212) NUMBER OF ROWS AND COLUMNS OF INPUT MATRICES AND
 NUMBER OF POLES TO BE INPUT FOR PLACEMENT
 EXAMPLE: NMU = 1 MEANS TO PLACE PCLE ONE AT A TIME
 CARD 7 A MATRIX ; READ BY ROWS AS:
 A(1,1) A(1,2)
 A(2,1) A(2,2)
 THERE WILL BE ONE OR MORE CARDS FOR EACH ROW OF A
 AFTER THE PRTOR CARDS ARE READ FOR A THE NEXT MATRIX MUST
 BE READ IN THE REMAINDER OF THIS COMMENT WILL REFER TO EACH NEW
 DATA SET AS A CARD. REMEMBER, THERE MAY BE SEVERAL CARDS PER
 SET SO THE NUMBERS DON'T CORRESPOND DIRECTLY WITH THE DATA
 LISTED IN THE INPUT FILE.
 CARD 8 E MATRIX; READ ROW BY ROW
 C MATRIX; READ ROW BY ROW

```

* CARD 10 F MATRIX; READ ROW BY ROW ENTERED HERE IS USED ONLY FOR SINGULAR VALUE
* ANALYSIS MODE, FOR OTHER MODES THE POLE PLACEMENT ROUTINE WILL GENERATE Q AND THEN F (FROM INNER ROUTINE)
* CARD 11 Q MATRIX; READ ROW BY ROW ( Q=0 WHEN NO STARTING VALUES IS USED )
* CARD 12 R MATRIX; READ ROW BY ROW FORMAT(8I2) THIS MATRIX DEFINES THE DESIGN VARIABLE
* CARD 13 IOW MATRIX; READ ROW BY ROW TO THE OPTIMIZER. THE ENTRY WOULD BE OF THE FORM:
*          1 2
*          0 3
* THIS WOULD TELL THE OPTIMIZER TO MAKE Q(1,1) THE FIRST DESIGN VARIABLE (X(1)) AND SO FORTH; THE ZERO TELLS THE OPTIMIZER THAT F(2,1) (IN THIS CASE) IS NOT A DESIGN VARIABLE AND IS NOT TO BE CHANGED. ONLY ONE OFF AS Q IS REQUIRED TO BE SYMMETRIC. ONLY ONE OFF DIAGONAL TERM IS REQUIRED; I.E. ONLY Q(1,2) OR Q(2,1) IS ASSIGNED. THE PROGRAM AUTOMATICALLY MAKES Q(2,1) ETC..
* CARD 14 VLB,VUB; FORMAT(2F10.0) THERE WILL BE ONE CARD FOR EACH DESIRED POLES
* CARD 15 REALMU,IMAGMU; FORMAT (2F10.0) INPUT REAL AND IMAGINARY DESIRED POLE LOCATION
*          ONE POLE PER CARD
* *****
* CALL ERFSSET(207,256,-1,1,209)
* CALL ERFSSET(215,256,-1,1)
* CALL ERFSSET(187,256,-1,1)
* *****
* ***** INPUT MODES
* ***** READ (1,750) KODE,KONTRL
* ***** WRITE (6,* ) 'KONTRLO',KONTRL
* *****
*--INPUT MAXN INITIAL FREQUENCY, DELTA FREQ, OPTIMIZER WEIGHTS-----
* READ (1,710) WMAX,W19,DELW
* READ (1,570) WT1,WT2,WT3
*--INPUT DESIRED SINGULAR VALUES (NOT USED) AND TYPE OF CONSTRAINTS---CCN02230
*          CCN02240
*          CCN02250
*          CCN02260
*          CON02270
*          CCN02280
*          CON02290
*          CCN02310
*          CCN02320
*          CON02330
*          CCN02340

```

```

READ (1,580) SMMINI,SMMIN0,RJ,NIDG
      N=19
      NC=1
C ----- INPUT OPTIMIZER INPUT DATA (SEE ADS MANUAL) -----
C
C     READ (1,760) IGRAD,NDV,NCON,ISTRAT,IOPT,IONED,IPRINT,INFO
C ----- INPUT THE PARAMETER VALUES AND THE MATRICES A,B,C,F,Qw,Rw-----
C
C     READ (1,770) NROWA,NCOLA,NROWB,NCOLE,NROWC,NCOLC,NROWF,NCOLF,
      INROWQ,NCOLQ,NROWR,NCOLR,NMU
      CALL READ ('A',NROWA,NCOLA)
      CALL READ ('B',NROWB,NCOLB)
      CALL READ ('C',NROWC,NCOLC)
      CALL READ ('F',NROWF,NCOLF)
      CALL READ ('QW',NROWQ,NCOLQ)
      CALL READ ('RW',NROWR,NCOLR)
      DO 13 I=1,NROWR
      DO 13 J=1,NCOLR
      RWSR(I,J)=Rw(I,J)
13    CONTINUE
C ----- SPECIFY THE DESIGN VARIABLES WITHIN THE Q MATRIX -----
C     IF FOR BOTH Q AND R IS TO BE VARIED, MODIFICATION
C     IS REQUIRED FOR THE FOLLOWING BLOCK
C
C     DO 10 J=1,NROWQ
      READ (1,780) IQW(J,K),K=1,NCOLQ
10    CONTINUE
C ----- SET THE DESIGN VARIABLE BOUNDS -----
C
C     DO 20 J=1,NDV
      READ (1,790) VLB(J),VUB(J)
20    CONTINUE
C ----- REAC THE DESIRED EIGENVALUES -----
C
C     DO 60 I=1,NMU
      READ (1,860) REALMU(I),IMAGMU(I)
      CONTINUE
60    DO 70 I=1,NMU
      MU(I)=CMPLX(REALMU(I),IMAGMU(I))
      CONTINUE
70    CONTINUE
C ----- SORT THE INPUT EIGENVALUES USING IMSL ROUTINE VSRTR -----
C
C     DO 80 I=1,NMU

```

IF (KODE.EQ.2.AND.KONTROL.EQ.2) GO TO 531

```

      WRITE (6,50)
      CALL CVECWR (MU,NMU)
C -----COMPUTE SORT NORMALISED THE EIG-VALUES AND VECTORS-----
C OF THE PLANT MATRIX A
      CALL EIGRF (A,NROWA,10,2,EIG,Z,10,MK,IER)
      DO 181 I=1,NROWA
      DO 183 J=1,NROWA
      WORK(J)=Z(J,I)
      CONTINUE
      SNCRM=SCNRM2(NROWA,WORK,1)
      SNCRM=1.0/SNCRM
      CALL CSSCAL(NROWA,SNCRM,WORK,1)
      DO 185 K=1,NROWA
      Z(K,1)=WCRK(K)
      CONTINUE
      185 CONTINUE
      181 CONTINUE

C-----BEGIN OF TRANSFORMATION BLOCK
C-----TRANSFORMATION MATRIX: 0Z• (REAL EIG) & 0Z*LX• (COMPLEX EIG)
C-----DO 190 J=1,NROWA
      REAL(EIG(J))=REAL(EIG(J))
      IEIG(J)=AIMAG(EIG(J))
C -----IF A HAS COMPLEX ROOT ICOMP=1
      IF (AIMAG(EIG(J)).NE.0.0) THEN
      ICMP=1
      ICIMP=1
      INITIALIZE LX MATRIX
      DO 197 K=1,NROWA
      DO 197 L=1,NCOLA
      LX(K,L)=0.0
      LX(K,K)=1.0
      CONTINUE
      CALL CPLEQU ( LX,LXI,NROWA,NCOLA)
      CALL CPLEQU ( LX,ZLI,NROWA,NCOLA)
      LX(1,1)=0.5
      LX(1,2)=0.0,-0.5
      LX(2,1)=0.5
      LX(2,2)=0.0,0.5
      CALL CPLEQU ( LX,TEMPX,NROWA,NCOLA)
      CALL LEGTIC(TEMPX,NROWA,10,LXI,NROWA,10,0,MKA,IER)
      197

```

```

C 190 CONTINUE
C-----END CF DC LOOP -----
      WRITE(6,*),**ICOMP ***,ICOMP
      WRITE(6,*),**AUX TRANSFORMATION L ***
      CALL CWRITE(LX,NROWA,NCOLA)
      DO 200 J=1,NROWA
      KEY(J)=J
      CALL VSRTR (REIG,NROWA,KEY)
      ***(VSRTING BY ALGEBRA OR ABS VALUE ****)
      CALL VSRTP (IEIG,NROWA,KEY)
      DO 220 J=1,NCOLA
      KEY(1:NCOLA+1-J)
      OEI(J)=EIG(K)
      DO 210 KK=1,NROWA
      OZ(KK,J)=Z(KK)
      TEMPX(KK,J)=Z(KK,K)
      210 CONTINUE
      220 CONTINUE
C-----ANY EIGENVECTOR ASSIGNMENT ROUTINE MAY BE INSERTED HERE IN PLACE OF
C-----THE EIGENVECTOR MATRIX OZ.
C-----WRITE (6,*),** EIGENVECTOR OF A ***
      CALL CPLREA(OZ,TEMPR,NROWA,NCOLA)
      CALL WRITE(TEMPR,NROWA,NCOLA)
      DO 222 I=1,NROWA
      DO 222 J=1,NCOLA
      OZ(I,J)=0.
      OZ(I,I)=1.0
      222 CONTINUE
      CALL CPLLEGU (OZ,OZ,TEMPX2,NROWA,NCOLA)
      CALL CPLLEGU (OZ,TEMPX2,NROWA,NCOLA)
      CALL CPLLEGU (TEMPX2,NROWA,1,0,OZ,I,NROWA,10,0,WA,IER)
      CALL CPLXCV (AX,A,NROWA,NCOLA)
      CALL CPLXCV (BX,B,NROWB,NCCLB)
      CALL CPLXCV (QWX,QW,NROWQ,NCOLQ)
C----- TRANSFORM AX (COMPLEX EIG VAULE )
C----- IF (ICOMP.EQ.1) THEN
      CONO3600
      CCNC3840
      CCNO3250
      CCNO2860
      CCNO3870
      CCNO3880
      CCNO3890
      CONC3500
      CONO32590
      CONO2590
      CONC3590

```

```

C      CALL CMATML (OZ,LX,NROWA,NCOLA,NROWA,TEMPX2)
C      CALL CMATML (AX,TEMPX2,NROWA,NCOLA,NROWA,TEMPX)
C      CALL CMATML (OZI,TEMPX,NROWA,NCOLA,NROWA,TEMPX2)
C      CALL CMATML (LXI,TEMPX2,NROWA,NCOLA,NROWA,AX)

C      ----- TRANSFORM BW -----
C      CALL CPLRQU (BX,TEMPX,NROWB,NCOLB)
C      CALL CMATML (OZ,LX,NROWA,NCOLA,NROWA,TEMPX2)
C      CALL LECTIC (TEMPX2,NROWA,10,ZLI,NROWA,10,O'WA,IER)
C      CALL CMATML (ZLI,TEMPX,NROWA,NCOLB,BX)

C      ----- TRANSFORM QWX -----
C      CALL CMATML (OZ,LX,NROWA,NCOLA,NROWA,TEMPX2)
C      CALL CMATML (QWX,TEMPX2,NROWQ,NCOLQ,NROWA,TEMPX)
C      CALL CMATML (TEMPX2,TEMPX,NROWC,NCOLQ,NROWA,QWX)

C      ELSE
C      ----- TRANSFORM AX (REAL EIG VALUE) -----
C      CALL CMATML (AX,OZ,NROWA,NCOLA,NROWA,TEMPX)
C      CALL CMATML (OZI,TEMPX,NROWA,NCOLA,NROWA,AX)

C      ----- TRANSFORM BX -----
C      CALL CPLRQU (BX,TEMPX,NROWB,NCOLB)
C      CALL CMATML (OZI,TEMPX2,NROWB,NCOLB,BX)

C      ----- TRANSFORM QWX -----
C      CALL CMATML (QWX,OZ,NROWQ,NCOLG,NROWA,TEMPX)
C      CALL CMATML (OZ,TEMPX,NROWQ,NCOLQ,NROWA,QWX)
C      END IF
C      ----- END OF IF THEN ELSE BLOCK -----
C      ----- OUTPUT TRANSFORMED A,B,Q -----
C      WRITE (*,*)
C      *** TRANSFORMED A MATRIX ***
C      CALL CPLRFA (AX,TEMPR,NROWA,NCOLA)
C      CALL WRITE (TEMPR,NROWA,NCOLA)
C      WRITE (*,*)
C      *** TRANSFORMED Q MATRIX ***
C      CALL CPLRFA (QNX,TEMPR,NROWG,NCOLQ)
C      CALL WRITE (TEMPR,NROWG,NCOLQ)
C      WRITE (*,*)
C      *** TRANSFORMED B MATRIX ***
C      CALL CPLRFA (BX,TEMPR,NRCWB,NCOLB)
C      CALL WRITE (TEMPR,NRCWB,NCOLB)

```

```

C ----- END CF TRANSFORMATION BLOCK -----
C -- FORMULATE THE DESIGN VARIABLE X FROM THE STARTING Q -----
C
C DO 50 J=1,NROWQ
C DO 40 K=1,NCULQ
C IF (IQ(J,K).EQ.0) GO TO 30
C   KK=IQ(J,K)
C   X(KK)=REAL(CWX(J,K))
C
C 30 CONTINUE
C 40 CONTINUE
C 50 CONTINUE
C----- CALL CPLXCV (RWX,RW,NROWR,NCOLR)
C----- CALCULATE R**-1/2 ----- USING LINPACK ROUTINE AND WURZEL ROUTINE
C----- CALL SPCCO(RW,10,NCCLR,RCOND,Z1,INFC1)
C----- CALL SPCCD1(RW,10,NCOLR,DET,1)
C DO 131 I=1,NROWR
C DO 131 J=1,NROWR
C   RW(J,I)=RW(I,J)
C 131 CONTINUE
C----- COMPUTE SQUARE ROOT OF MATRIX R
C CALL WURZEL(RW,TEMPR,NROWR,0.9,1.E-6,10)
C CALL CPLXCV (RWSSQR,TEMPR,NROWR,NCOLR)
C WRITE (*,*)
C CALL WRITE(TEMPR,NRCWR,NROWR)
C----- **** ENTER OPTIMIZATION BLOCK HERE***** 
C IF KCNTRL SET GREATER THAN 0 THE FIRST PASS JUMPS THE OPTIMIZER
C IF (KCNTRL.GT.0) GO TO 130
C----- CALL THE ADS OPTIMIZER TO SOLVE FOR THE BEST Q'S
C----- THIS FIRST CALL SETS THE SCALE FLAG SEE ADS MANUAL
C INFO=-2
C CALL AES (INFO,ISTRAT,IOPT,IONED,IPRINT,IGRAD,NDV,NCON,X,VLB,VUB,0
C 1BJ,G,IGD,NGT,IC,DF,XA,NRA,NCA,WKI,NRWK,IWK,NRIWK)
C----- CON02610
C----- CON02620
C----- CON02630
C----- CON02640
C----- CON02650
C----- CON02660
C----- CON02670
C----- CON02680
C----- CON02690
C----- CON02700
C----- CON02600
C----- CON03170
C----- CON03180
C----- CON03190
C----- CON03200
C----- CON032100
C----- CON032200
C----- CON03230
C----- CON03240
C----- CON032100
C----- CON03200
C----- CON03270
C----- CON03280
C----- CON03290
C----- CON03300

```

```

WK(3)=-C.05      IMPLIES NO SCALING -
C   -  WK(2)=0
C   -  WK(2)=0
C
C THIS CALL STARTS THE OPTIMIZATION PROCESS
C
C 120 CALL ADS (INFO,ISTRAT,IOPT,IONED,IPRINT,IGRAD,NDV,NCON,X,VLE,VUB,0
C        1BJ,G,LEG,NGT,IC,DF,IXA,NRA,NCA,WKI,NRWK,WK,NRWK)
C        IF ((INFC.EQ.0)) GO TO 480
C        IF ((INFC.GT.1)) GO TO 470
C
C THIS PORTION OF THE PROGRAM DOES THE ACTUAL COMPUTATION
C
C 130 DO 160 J=1,NROWQ
C        DO 150 K=1,NCOLQ
C
C-----EQUATE DESIGN VECTOR X WITH THE DESIRED SYSTEM MATRIX -----
C
C 140 IF ((IQW(J,K).EQ.0)) GO TO 140
C        KK=IQW(J,K)
C        QWX(K,J)=X(KK)
C        QWX(J,K)=X(KK)
C        CONTINUE
C
C 150 CONTINUE
C
C 160 CONTINUE
C
C CALL CPLXCV {CX,C,NRCWC,NCOLC}
C CALL CPLXCV {FX,F,NROWF,NCOLF}
C
C MAIN LOOP TO COMPUTE G(S) WHERE G=G(S)-A*(-1)*B-
C IT BE USED IN THE OPTIMIZER'S OBJECTIVE FUNCTION
C
C DO 263 I=1,NMU
C
C----- PLANT1 COMPUTE G(S) WHERE G=G(S)-A*(-1)*B-
C WHERE S IS THE DESIRED POLE LOCATION
C CALL PLANT1(GPS,AX,BX,CX,MU(I),NROWA,NCOLA,NROWB,NCOLB,NROWC,
C           NCCLC)
C----- PLANT1 COMPUTE G(S) WHERE G=G(S)-A*(-1)*B-
C WHERE S IS THE DESIRED POLE LOCATION

```

```

CALL PLANTI(GMS,AX,BX,CX,-MUL),NRONA,NCOLA,NROWB,NRQWC,
1 NCOLC
CALL CNATML(GPS,RNSGX,NROWA,NCOLA,NCOLR,GPSRSQ)
CALL CNATML(GMS,GPSASQ,NROWA,NCOLQ,NCOLR,GQR)
CALL CNATML(GMS,GQR,NROWA,NCOLA,NCOLR,GQR)
CALL CNATML(GMS,GXR,NROWA,NCOLR,NCOLR,GQR)
C
C      I+R**-(1/2)*IC(-S)*TQI(S) I R**-(1/2) II
C
C      DO 914 J=1,NRUWR
C      DO 913 K=1,NCOLR
C      XX1(J,K)=C*0
C      XX1(J,J)=1.0
C      XIRQR(J,K)=XX1(J,K)+RSGCR(J,K)
C
913 CNTINUE
C      ----- FINDING DETERMINANT TO BE USED IN THE OBJECTIVE FUNCTION ----
C
C      CALL CGED(XIRQR,10,NCOLR,IPVT,PCOND,WRK),
C      CALL CGED(XIRQR,10,NCOLR,IPVT,PCOND,WRK)
C      DETERM(1)=DETERM(1)*REAL(DET(1))
C
C53 CNTINUE
C      ----- END OF DO LOOP -----
C
C      -----FCRM QUANTITIES TO USE IN THE COST CRITERIA OF OPTIMIZER-----
C      EACH REAL PCLE REQUIRED ONE COST, EACH COMPLEX PAIR PDL REQUERED
C      TWO COST(COST1 AND COST2)
C
C      COST1=((REAL(DETERM(1))**2+AIMAG(DETERM(1))**2)
C      COST2=((REAL(DETERM(2))**2+AIMAG(DETERM(2))**2)
C      COST3=((REAL(DETERM(3))**2+AIMAG(DETERM(3))**2)
C      WRITE(6,* ) 'X',COST1,DETERM(1),X(1)
C
C      -----THE OBJECTIVE FUNCTION IS INSERTED HERE-----
C      OBJ=COST1+COST2+COST1*COST2
C
C      WRITE(6,* ) 'OBJ'
C      ----- CONSTRAINT EQUATION (IDG(3)=-2 FOR COMPLEX ROOTS) -----
C      IF (NCCH.EQ.0) GO TO 460
C      DO 450 J=1,2
C          IDG(J)=NIDG
C      CNTINUE
C      IDG(3)=-2
C
450 C-----OTHER CONSTRAINTS CAN BE INTRODUCED HERE-----

```



```

      WRITE (6,*)
      CALL WRITE (Q,NROWQ,NCOLQ)
      CALL SINGULAR VALUE ANALYSIS ONLY
      IF (KODE.EQ.2.AND.KCTRLE.Q.2) GO TO 532
      N2 = 2*NCOLA
      *****
      ---- THE REDUCE ROUTINE MATCHES THE VARIABLE FROM THE CPTPP
      ---- WITH THE MODIFIED VERSION OF THE INNER ROUTINE FROM
      CPTSYS PROGRAM
      ----
      THE SYSTEM MATRIX A,B,C, AND Q, FROM CPTPP, ARE USED
      BY THE INNER ROUTINE TO COMPUTE THE FEEDBACK GAIN F.
      F CAN THEN BE USED IN THE SINGULAR VALUE ANALYSIS PORTION OF
      THIS PROGRAM
      ----
      CALL REDUCE (A,B,C,Q,RWSP,DUMA,DUMB,DUMC,DUMQ,DUMR,DUMRI,
      1,NRCWA,NCCLB)
      ----
      CALL INNER (NCOLA,NCCLE,NRCWC,N2,DUMA,DUMB,DUMC,DUMQ,DUMR,FBCG
      1,K'AA,FRC,XLGI,DUMRI)
      DO 533 I=1,NROWF
      DO 533 J=1,NCOLF
      F(I,J)=-FBCG(I,J)
      533 CONTINUE
      CALL CPLXCV (AX,A,NRCWA,NCOLA)
      CALL CPLXCV (BX,B,NRCWB,NCCLB)
      CALL CPLXCV (CX,C,NRCWC,NCCLC)
      CALL CPLXCV (FX,F,NRCWF,NCCLF)
      ---- THE FINAL FEEDBACK GAIN MATRIX IS DISPLAY HERE
      ----
      WRITE (6,840)
      CALL WRITE (F,NROWF,NCOLF)
      ----
      COMPUTE THE A-B*C SYSTEM MATRIX FOR THE NEXT PGLE ASSIGNMENT
      ----
      CALL MMUL (F,C,NROWF,NCOLF,NCOLC,FC)
      CALL MMUL (B,FC,NRCWE,NCULE,NCOLC,BFC)
      DO 180 I=1,NROWA
      DO 170 J=1,NCOLA
      CONO3130
      CONO3140
      CONO3E30
      CONO3E40
      CONO2E50
      CONO2E60
      CONO3E70
      CONO3E80
      CONO3E90
      *****

```

```

170 AMBFC(I,J)=A(I,J)-BFC(I,J)
180 CONTINUE
C WRITE(6,890)
C -----OUTPUT (A-BFC)
C -----CALL WRITE (AMBFC, NRROWA, NCOLA)
C
C IF (KODE.EQ.0) GO TO 555
C **** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C -----SINGULAR VALUE ANALYSIS BEGIN HERE
C
C CALL THE PLANT TRANSFER MATRIX ROUTINE AND FORM THE SYSTEM
C RETURN DIFFERENCE MATRICES AS REQUIRED
C
C DEL=DELM
C I=W19
C NINT=1
C M=MINAI=C.
C MINAI=C.
C MINAO=0.
C MINMO=C.
C MAXAI=C.
C MAXM=C.
C MAXAC=C.
C MAXI=C.
C PLXCV (AX,A,NROWA,NCOLA)
C PLXCV (BX,B,NROWB,NCOLB)
C PLXCV (CX,C,NROWC,NCOLC)
C PLANT (PLAN,PLX,MAXI,NROWA,NCOLA,NROWB,NCOLB,NCOLC)
C CNTML (FX,PLAN,NROWF,NCOLF,NCOLB,NROWC,NCOLC,PLTFX)
C CALL CNATML (PLAN,FX,NROWF,NCOLF,NCOLB,NROWC,NCOLC)
DO 280 J=1,NROWF
DO 270 K=1,NCOLB
  XI(J,K)=0.0
270 CONTINUE
280 DO 290 J=1,NROWF
  XI(J,J)=1.0
290 DO 310 J=1,NROWC
  DO 300 K=1,NCOLF
    DO 300 K=1,NCOLB
      XX1(J,K)=0.0
      XX1(J,J)=1.0
      XI(PFFX(J,J),K)=XX1(J,K)+PLTFX(J,K)
300 CONTINUE
DO 330 J=1,NROWC

```

```

DO 320 K=1,NCOLB
XI PXP(J,K)=XI(J,K)+FXPLT(J,K)
CONTINUE
CALL CPLEGU(XIPXP,TEMPX,NROWF,NROWF,NROWF)
DO 340 K=1,NROWF
DO 350 J=1,NROWF
FX PLTI(J,K)=0.0
FX PLTI(J,J)=1.0
CONTINUE
DO 370 J=1,NROWC
DO 380 K=1,NROWC
PLTFXI(J,J)=J.0
PLTFXI(J,J)=1.
CONTINUE
C THIS ESTABLISHES THE RETURN DIFFERENCE MATRIX FOR MULT. CASE
C CALL LECTIC(FXPPLT,NROWF,10,FXPLTI,NROWF,10,0,WA,IER)
C CALL LECTIC(PLTFX,NROWC,10,PLTFXI,NROWC,10,0,WA,IER)
DO 390 J=1,NROWF
DO 380 K=1,NCOLB
XI PLTI(J,K)=XI(J,K)+FXPLTI(J,K)
CONTINUE
DO 410 J=1,NROWC
DO 400 K=1,NROWC
XI PLTI(J,K)=XXI(J,K)+PLTFXI(J,K)
CONTINUE
CONTINUE
DO 390
DO 410
CONTINUE
CONTINUE
C DO SINGULAR VALUE DECOMP AND QUANTIFY ALL DESIRED SV'S
C----- THIS PORTION COMPUTES THE MAXIMUM EIG-VALUES OF THE -----
C----- RETURN DIFFERENCE MATRIX WHICH GIVE THE UPPER BOUND
FOR THE SINGULAR VALUE
C----- CALL EIGCC(TEMPX,NROWF,10,0,EIG,Z,IC,WK,IER)
DU 411 I=1,NROWF
ME1G(I)=SQR((REAL(EIG(I)))*2+(AIMAG(EIG(I)))*2)
IF (EIG(I)) THEN
  ME1G(I)=ME1G(I)

```

```

      IF (MEIG(I)*LT.MEIG(I-1)) THEN
        MEIG(I)=MEIG(I-1)
      END IF
      CALL YSFIM (MEIG(NROWF))
      MEIG(ACNT)=MEIG(NROWF)

      ONEGA(ACNT)=W
      SIGNM1(ACNT)=SV(NROWF)
      SIGNM2(ACNT)=SV(NRCWF-1)
      SIGNM3(ACNT)=SV(NRCWF-1)
      SVADM(ACNT)=SVAC(NROWC)
      SVACX(ACNT)=SVAC(1)
      SVRIM(ACNT)=SVRI(NROWF)
      SVRMIX(ACNT)=SVRI(1)
      SYMMO(ACNT)=SYMO(NRCWF)
      SYMMX(ACNT)=SYMM(1)
      SGNPRO(ACNT)=SQRT(SIGNM1(NCNT)*SIGNM2(NCNT))
      SMINA1=(SMINAI+(AMAXI(0.0,SMIN1-1))/2)
      SMINA0=(SMINA0+(AMAXI(0.0,SMIN0-1))/2)

      C--CCMPUTE THE OPEN LOOP TRANSFER FUNC
      C--FOR TWO INPUT SYSTEM ONLY-NEED AUDI
      C
      CALL PLANT(TMPX,AX,BX,FX,w,NRCWA
      INCOLF)
      LP11(ACNT)=10.*ALOG10((REAL(TEMPX
      1+0.20000C)
      LP12(ACNT)=10.*ALOG10((REAL(TEMPX
      1+0.00000J)
      LP21(ACNT)=10.*ALOG10((REAL(TEMPX
      1+0.00000C)
      LP22(ACNT)=10.*ALOG10((REAL(TEMPX
      1+0.00000C))

      C--CCMPUTE LCCP FOR ALL DESIRED FREQUENC
      C
      K=w+DEL
      wD=10.0*w/I
      IF (wLT.wD) GO TO 420
      DEL=10.0*C*DEI
      WI=10.0*C*DEI
      NCNT=NCNT+1

```

```

IF (W.LE.WMAX) GO TO 260
MAX=NCAJ-1

C----- OUTPUT THE OPTIMIZED SINGULAR VALUE DATA -----
C
C      WRITE (6,73C)
C      WRITE (3,60C) MAX
DO 540 J=1,MAX
C      WRITE (2,67C) CMESA(J),SIGNM1(J),SIGNM2(J),SIGNMX(J),SIGPRO(J)
C      WRITE (2,74C) CMESA(J),SIGNM1(J),SIGNEIG(J),SIGNMX(J),SIGPRE(J)
C      WRITE (2,68C)
C      DO 555 C J=1,MAXN
C      WRITE (2,69C) SVADM0(J),SVMM0(J),SVMMX(J),SVMM0(J),SVMMX(J)
C      WRITE (2,70C) CMEGA(J),SIGNM(J),PLP22(J),LPM(J),SVMMX(J)
C      WRITE (2,71C) SVXG(J),SVADM0(J),SVMM0(J),SVMMX(J),SVMMX(J)
C      IF (KONTRL.EQ.0.OR.KONTROL.EQ.1) GO TO 560
C      WRITE (6,*)
C      KONTROL=C
C      GO TO 12C
C      WRITE (6,*) *END OF OPTPP ANALYSIS *
C***** **** **** **** **** **** **** **** **** **** **** **** ****
C----- FORMAT STATEMENTS -----
C
C      560 FORMAT (STCP)
C      570 FORMAT (F10.0)
C      580 FORMAT (F10.0,15)
C      590 FORMAT (/,10H WEIGHT 1=F10.5,9HWEIGHT 2=F10.5,9HWEIGHT 3=F10.5,
C      1/   FFORMAT (/,9H SVMINI=,F10.5,8H SVMINO=,F10.5,5H RJ=,F10.5,6H
C      600 I=15,/)
C      610 FFORMAT (215,1
C      620 FFORMAT (/,28H *** THE INITIAL DATA IS *** '/)
C      630 FFORMAT (/,41H *** THE OPTIMIZER COMPUTED OUTPUT IS *** ,/ )
C      640 FFORMAT (2E12.5) *** THE OBJECTIVE FUNCTION IS =,F10.5,6H
C      650 FFORMAT (/,34H *** THE OBJECTIVE FUNCTION IS =,F10.5,6H
C      660 FFORMAT (5E12.5)
C      670 FFORMAT (/,2X,7H FREQ,5X,6HSVADMO,4X,5HSVADX0,4X,5HSVMM,4X,5HSVM
C      680 IX,4X,5HSVMM0,4X,5HSVMMX,/) *****,/ )
C      690 FFORMAT (6E12.5)
C      700 FFORMAT (7F10.5)
C      710 FFORMAT (3F10.0)
C      720 FFORMAT (/,6H WMAX=,F10.5,1X,10HINIT FREQ=,F10.5,1X,10HFREQ STEP=,
C      1/ 10.5,1/ )
C      730 FFORMAT (/,10H FREQUENCY,5X,13HMIN ADD IN SV,4X,13H2ND ADD IN SV,3CUNO5560

```

```

1X,13HMAX ADC IN SV),2X,18HMIN PROD ADD IN SV)
FORMAT (6F10.5,1X)
FORMAT (15,15)
FORMAT (1312)
FORMAT (1512)
FORMAT (2F10.0)
FORMAT (1H1,7/1,12H<<< RUN #,15,6H >>>,///)
FORMAT (27H ***,THE A PLANT MATRIX ***,//)
FORMAT (//,35H ***,THE B CONTROL INPUT MATRIX ***,//)
FORMAT (//,33H ***,THE C OBSERVATION MATRIX ***,//)
FORMAT (//,33H ***,THE D WEIGHTING MATRIX ***,//)
FORMAT (//,33H ***,THE E DESIGN VARIABLE (S) X ***,//)
FORMAT (//,38H ***,THE F MATRIX ***,//)
FORMAT (//,43H ***,THE G WEIGHTING MATRIX ***,//)
FORMAT (//,44H ***,THE H COMPLEX EIGENVALUES (INPUT),//)
FORMAT (2F10.0) **** THE ORDERED COMPUTED EIGENVALUES,//
FORMAT (//,37H ***,THE OPTIMAL Q MATRIX //)
FORMAT (//,29H ***,THE AUGMENTED A-E*F*C MATRIX ***,//)
FORMAT (//,37H ***,THE AUGMENTED (8FIC.4) )
FORMAT (8FIC.4)
END ****
C ****
C **** THE END OF THE MAIN PROGRAM (OPTPP) ****
C ****

```

CONSIDE THE STANDARD IMSL ROUTINES, THE FOLLOWINGS ARE REQUIRED:
CEISPACK ROUTINES:

LINPACK FCTNSES: SPOOC, SPOOI, CGECO, CGEDI


```

DU 370 K=1,NC
PR C(I,J)=PRC(I,J)+BL1(I,K)*G1(J,K)
C94 I=NLE=1,NH
DO 380 J=1,NH
RM(I,J+NH)=C*0
DO 380 K=1,NC
RM(I,J+NH)=RM(I,J+NH)-GL(I,J+NH)-GL(I,K)*PRC(K,J))
CONTINUE
C-----2NX2N HAMILTONIAN MATRIX---M11 AND M22-----
C-----DIAGONAL BLOCKS---M11 AND M22-----
C-----KRITE (t,*)
C-----CALL WRTE(BA,2,2)
C-----DO 390 I=1,NH
C-----DO 390 J=1,NH
RM(I,J)=C*EC
RM(I,J)=BA(I,J)
RM(I+NH,J+NH)=BA(J,I)
RM(I+NH,J)=RM(I+NH,J)
CONTINUE
C-----M12 BLOCK IS DEFINED IN LINE 430 ABOVE-----
C-----M12 BLOCK IS DEFINED IN LINE 430 ABOVE-----
C-----CONTINUE
C-----CALL EISPACK ROUTINES-----
C-----410 CALL BALANC(M,M,MLOW,IHIGH,D1)
C-----CALL ORTHES(M,M,MLOW,IHIGH,RM,D2,X1)
C-----CALL CRTRAN(M,M,MLOW,IHIGH,RM,W1,X1,IERR)
C-----WRITE(t,*),IERR
IF (IERR .NE. 0) CALL EXIT(M,RM,IERR)
CALL BALEAK(M,M,MLOW,IHIGH,D1,X1)
C-----DEBUG DIAGNOSTICS ON EULER-LAGRANGE EQUATIONS---
430 CONTINUE
IF (NOB.EQ.0) WRITE(6,1250)
C-----CALL RGM(1,NS,NC,NCB,WR,MI,X1,GN,W1,GR,CHI,SC,WHS,
1D2)
CONTINUE
450 CONTINUE
470 CONTINUE
C-----CALCULATION OF FEEDBACK GAIN---U=(BINVERSE)*GT*G16
C-----FEEDBACK GAINS---CALCULATE GT
C-----DO 480 I=1,NS
DO 480 J=1,NS
PR C(I,J)=PRC(I,J)+GL(I,J)*GN(K,J)
DO 490 I=1,NC

```

```

      DD 490 J=1,NS
      DD 490 J=0,NC
      FBGC(I,J)=FBGC(I,J)-BLI(I,K)*PRO(K,J)
C-----NORMALIZE AND PRINT OPT. REG. CLOSED LOOP EIGENSYSTEM
      IWRITE=2
      CALL CNGRM(CWR,CWI,SC,NS,WRITE,NSC,DDD,D1,D2,WNORM,WNCRM,FBGC,
      1AA,NC,NS)
C-----THE OPTIMUM FEEDBACK CONTROL GAINS
      C500 CONTINUE
      RETURN
C-----FORMAT //(*5X,45HOPEN LOOP DYNAMICS MATRIX.....F...,//)
      1360 FORMAT(10(2X,0PD11.4))
      1290 FORMAT(10(5X,45HTHE CONTROL DISTRIBUTION MATRIX.....G...,//)
      1400 FORMAT(10(5X,45HTHE CONTROL COST MATRIX.....B...,//)
      1410 FORMAT(10(5X,45HMEASUREMENT SCALING MATRIX.....P...,//)
      1440 FORMAT(10(5X,45HUT COST MATRIX.....F...,//)
      1460 FORMAT(10(5X,28H*DESTABILIZATION CASE.....A...,//)
      1480 FORMAT(10(5X,28H*ADDED DOWN *10X,39HTHE ECULLO CPT113180
      1KING VALUES WILL BE *10X,49HTHE DIAGONAL OF THE "F" MACPT113190
      2TP IX SYSTEM DESTABILIZE IT *10X,41H OPTIMAL GAINS FOR THE STABILIZE CPT113200
      2D SYSTEM, SYSTEM CALCULATIONS *10X,39H ARE THEN USED AS FIXED SUBOPTIMAL GAINS, *10X,28CPT113210
      4HF CRATE //43H PROGRAM TERMINATING DUE TO UNSTABLE SYSTEM
      1490 FORMAT(10(5X,31HOPEN LOOP TRANSFER FUNCTION SYSTEM)
      1500 FORMAT(10(5X,32H FULFILER-LAGRANGE SYSTEM MATRIX *//)
      1510 FORMAT(10(5X,43HE EIGENVALUES AND EIGENVECTORS OF THE 2N X 2N,/,5X,
      1520 FULFILER-LAGRANGE SYSTEM AFTER QR2.....//)
      1530 FORMAT(1X,1P,2E13.6)
      1540 FORMAT(1X) CPT113230
      1550 FORMAT(1X,5X,41HE LICENSE SYSTEM OF OPTIMAL REGULATOR.....//)
      1560 FORMAT(1X,5X,41HE LICENSE SYSTEM OF OPTIMAL ESTIMATOR.....//)
      1570 FORMAT(1X,39HE EIGENVECTORS FROM REGAIN PRIOR TO CNOIK.....//)
      1580 FORMAT(1X,2X,57HTHE OPTIMAL FEEDBACK GAIN CONTROL MATRIX...C=B INV CPT113240
      1*GST*..*(10(1X,1P,E1.4))
      1590 FORMAT(10(2X,1PE11.4))
      1600 FORMAT(10(5X,45HTHE CLOSELY LOCP DYNAMICS MATRIX *//)
      1610 SYSTEM) CPT113250
      1640 FORMAT(1X,2X,1P,6D14.6) CPT113260
      1690 FORMAT((5(1X,1P,D1.6))) CPT113270
      1750 FORMAT(1X,1PD15.7,25X,D15.7) CPT113280
      2000 FORMAT(5X,2E13.6) CPT113290
C-----END

```



```

10      U(I,J)=U(I,J)+R(I,INDEX)*T(INDEX,J)
10      CONTINUE
30      CONTINUE
50      RETURN
END
C ****
C
C-----CSVD ALGORITHM FOR COMPLEX MATRICES; ANY OF THE ROUTINES OF IMSL
C-----CR LINPACK COULD ALSO BE USED WITH PROPER CAUTIONS
C-----SUBROUTINE CSVD (A,MMAX,NMAX,M,NP,NU,NV,S,U,V)
C-----COMPLEX AL(MMAX,1),U(NMAX,1),V(NMAX,1)
C-----INTEGER M,N,P,NU,NV
C-----REAL S(1)
C-----COMPLEX Z,R
C-----REAL BTA,C,C,T(100),T(100)
C-----DATA BTA,TOL/1.5E-8,I.E-31/
C-----NP=N+P
N1=N+1

C      USE HOLLOW REDUCTION
C      C=0.E0
K1=K+1
10      K1=K+1
      C      ELIMINATION OF A(I,K), I=K+1, . . . , M
      Z=C.E0
      DO 20  I=K.M
      Z=Z+R*REAL(A(I,K))*2+AIMAG(A(I,K))*2
      B(K)=C.E0
      IF ((Z.LT.TOL) GO TO 70
      Z=SGRT(Z)
      B(K)=Z
      W=CABS(A(K,K))
      Q=C.L.EC.C.EC
      IF ((W.NE.0.E0) Q=A(K,K)/W
      A(K,K)=C*(Z+W)
      IF ((K.EG.NP) GO TO 70
      DO 50  J=K1.NP
      Q=0.E0
      DO 30  I=K.M
      Q=Q+C*CNS((A(I,K)).NE.A(I,J))
      G=G/(Z*(Z+W))
      DO 40  I=K.M
      A(I,J)=A(I,J)-Q*A(I,K)
      CONTINUE
C
C-----CWRCC640
C-----CWRCC650
C-----CWRCC660
C-----CWRCC670
C-----CWRCC680
C-----CWRCC690
C-----CWRCC700
C-----CWRCC710
C-----CWRCC720
C-----CWRCC740
C-----CWRCC750
C-----CWRCC760
C-----CWRCC770
C-----CWRCC780
C-----CWRCC790
C-----CWRCC800
C-----CWRCC810
C-----CWRCC820
C-----CWRCC830
C-----CWRCC840
C-----CWRCC850
C-----CWRCC860
C-----CWRCC870
C-----CWRCC880
C-----CWRCC890
C-----CWRCC900
C-----CWRCC910
C-----CWRCC920
C-----CWRCC930
C-----CWRCC940
C-----CWRCC950
C-----CWRCC960
C-----CWRCC970
C-----CWRCC980
C-----CWRCC990
C-----CWRCC1000
C-----CWRCC1010
C-----CWRCC1020
C-----CWRCC1030
C-----CWRCC1040
C-----CWRCC1050
C-----CWRCC1060
C-----CWRCC1070
C-----CWRCC1080
C-----CWRCC1090

```

```

C PHASE TRANSFORMATION OR A(K,J) /CAES(A(K,K))
C Q=-C(J,K) DO J=K,NP
C DO EO J=K,NP A(K,J)=C*A(K,J)
C
C 70 ELIMINATION OF A(K,J) DO J=K+2,...,N
C IF (K.EQ.N) GO TO 140
C Z=C.EQ.0 DO 80 J=K1,N
C Z=Z+REAL(A(K,J))**2+AIMAG(A(K,J))**2
C C(K1)=C.EQ.0
C IF (Z.LT.TOL) GO TO 130
C Z=SQR(Z)
C C(K1)=Z
C W=CABS(A(K,K1))
C E=C(1.+NE.C.EQ.0.EC)
C IF (W.NE.0.C.EQ.0) Q=A(K,K1)/W
C A(K,K1)=C*(Z+M)
C DO 110 I=K1,M
C Q=(C.EQ.0.C.EQ.0)
C DO 90 J=K1,N
C U=G+CCNJC(A(K,J))*A(I,J)
C Q=G/(Z*(Z+M))
C DO 100 J=K1,N
C A(I,J)=A(I,J)-Q*A(K,J)
C C110 NLE
C
C 90 PHASE TRANSFORMATION
C Q=-C(JNJC(A(K,K1))/CABS(A(K,K1)))
C DO 120 I=K1,M
C A(I,K1)=A(I,K1)*Q
C K=K1
C GO TO 100
C
C 100 TOLERANCE FOR NEGIGIBLE ELEMENTS
C EP.S=0.EO
C DO 140 K=1,N
C S(K)=E(K)
C T(K)=C(K)
C EPS=AMAX1(EPS,S(K)+T(K))
C EPS=EPS*.ETA
C
C 120 IF (NU.EG.0) GO TO 180
C DO 130 J=1,NU
C DO 160 I=1,M
C U(I,J)=(C.EC,0.EO)
C
C 130
C 140
C 150
C
C 160

```

```

170 U(J,J)=(1.EC,0.E0) GO TO 210
170 IF (NV*0)=1, NV
DO 200 J=1,N
DO 190 I=1,N
V(I,J)=(0.EC,0.E0)
V(J,J)=(1.EC,0.E0)
C QR DIAGONALIZATION
190 DO 280 KK=1,N
K=N1-KK

C TEST FOR SPLIT
200 DO 230 LL=1,K
L=K+1-L
IF (ABS(T(L))-LF.EPS) GO TO 290
IF (ABS(S(L-1)).LE.EPS) GO TO 240
CONTINUE

C CANCELLATION OF E(L)
230
CS=C*EC
SN=1.*EO
L1=L-1
DO 280 I=L,K
F= SN*T(I)
T(I)=CS*T(I)
IF (AD(S(F)).LE.EPS) GO TO 290
H=S(I)
W=SQR(T(F*F+T**H))
S(I)=W
CS=h/W
SN=-F/W
IF (NU.EQ.0) GO TO 260
DO 250 J=1,N
X=REAL(U(J,L1))
Y=REAL(U(J,I))
U(J,L1)=CMPLX(X*CS+Y*SN,0.E0)
U(J,I)=CMPLX(Y*CS-X*SN,0.E0)
250 IF (NP.EC.N) GO TO 280
DO 270 J=N1, NP
Q=A(U,J)
R=A(I,J)
A(I,J)=C*CS+R*SN
A(I,J)=R*CS-Q*SN
CONTINUE

C TEST FOR CONVERGENCE
270 IF (L.EC.K) GO TO 360

```



```

Q= A(I-1,J)
R= A(I,J)
A(I-1,J)=6.*CS+R*S
A(I,J)=R*C-S-Q*S
CUNTINLE
T(L)=O.E0
T(K)=F
S(K)=X.220
GJ TO 220
C CONVERGENCE
340 IF (W.GE.0.E0) GO TO 380
S(K)=-K
IF (INV.E=0) GO TO 380
DO 370 J=1,N
V(J,K)=-V(J,K)
CUNTINLE
350
360 SCFT SINGULAR VALUES
DO 450 K=1,N
370 J=K
DO 390 I=K,N
IF (S(I).LE.6) GO TO 390
G=S(I)
J=I
CONTINUE
IF (U(J,E.K)) GO TO 450
S(J)=S(K)
IF (INV.E=0) DO 400 I=1,N
DO 400 I=1,N
V(I,J)=V(I,K)
400 IF (INU.EG.0) GO TO 410
DO 420 I=1,N
G=U(I,J)
U(I,J)=U(I,K)
420 U(I,K)=G.0
IF (NU.EG.0) GO TO 430
430 DO 440 I=N1,NP
Q=A(I,1)
A(J,I)=A(K,I)
A(K,I)=G
CONTINUE
440
450
C BACK TRANSFORMATION

```



```

20 FORMAT (5F12.0)
END
C *****
C----- WRITE A REAL MATRIX
C----- SUBROUTINE WRITE (A,N,K)
REAL*4 A(10,10)
INTEGER N,K
DO 10 I=1,N
    WRITE (6,20) (A(I,J), J=1,K)
10 CONTINUE
RETURN
FORMAT (5F12.5)
END
C *****
C----- CONVERT REAL MATRIX TO COMPLEX FORM
C----- SUBROUTINE CPLXCV (X,A,N,K)
COMPLEX *8 X(10,10)
REAL*4 A(10,10)
INTEGER N,K
DO 20 I=1,N
    DO 10 J=1,K
        X(I,J)=COMPLEX(A(I,J),0.00)
10 CONTINUE
20 RETURN
END
C *****
C----- CONVERT REAL PART OF COMPLEX MATRIX TO REAL MATRIX
C----- SUBROUTINE CPLRFA (X,A,N,K)
COMPLEX *8 X(10,10)
REAL*4 A(10,10)
INTEGER N,K
DO 20 I=1,N
    DO 10 J=1,K
        A(I,J)=REAL(X(I,J))
10 CONTINUE
20 RETURN
END
C *****

```

EQUATIONS IN COMPLEX MATRIX THEORY

CHP02630

```

C SUBROUTINE CPLFQU (X,Y,N,K)
C COMPLEX *8 X(10,10),Y(10,10)
C REAL*4 A(10,10)
C INTEGER I,J,N,K
C DO 10 I=1,N
C DO 10 J=1,K
C Y(I,J)=X(I,J)
C CONTINUE
C END

```

THIS ROUTINE WILL COMPUTE THE TRANSFER FUNCTION AS A
FUNCTION OF FREQUENCY FOR A GIVEN A, B, C SET OF
MATRICES

```

SUBROUTINE PLANT (GP, AX, BX, CX, NFA, NCA, NRB, NCB, NRC, NCC)
COMPLEX #8CX(10,10), AX(10,10), BX(10,10), GP(10,10), I(10,10),
        SIAI(10,10), SI(10,10)
        COMPLEX S
        REAL*4 hA(100)
        S=CMPLX(C,0.0)
        DO 20 J=1,NRA
        DO 10 K=1,NCA
        SI(J,K)=C.0
        SI(J,K)=C.0
CONTINUE
        DO 30 J=1,NCA
        SI(J,J)=1.0
        SI(J,J)=SI(J,J)
        DO 50 J=1,NRA
        DO 40 K=1,NCA
        SI(AJ,K)=SI(J,K)-AX(J,K)
CONTINUE
        CALL LECTIC (SIA, NCA, 10, 1, NCA, 10, 0, "A, 1ER)
        CALL CHATML ((1,BX,NCA, NCA, NCB, SIAIB,
        CALL CHATML (CX, SIAIE, NRC, NCC, NCB, GP)
RETURN

```

THIS ROUTINE IS INPUT FOR DE SULLVER
WHEN DE SULLVER IS ADDED DIRECTLY TO CONSV

CWR04C50
CWR04C60


```

20      CONTINUE
DO 30 J=1,NCA
   I(J,J)=I*1(J,J)
30      SI(J,J)=I*(J,J)
DO 50 J=1,NFA
   D(J,J)=SI(J,J)
50      DO 40 K=1,NCA
      SI(I,J,K)=SI(J,K)-AX(J,K)
40      WRITE(*,*) 'SI='
      CALL CNECITC(SIA,NRCWA,NCOLA)
      CALL CMATML(SIA,NCA,10,I,NCA,10,O,A,IER)
      CALL CMATML(SIA,NCA,NCA,NCA,NCA,NCA,NCA,NCA)
      CALL CMATML(SIA,NRC,NCC,NCC,NCB,NCB,GP)
      RETURN
END
* * * * *
C----- MATRIX MULTIPLICATION WITH COMPLEX MATRIX(A*T*B)
C----- SUBROUTINE CMATML(R,T,M,LL,N,U)
C----- COMPLEX X*8R(10,10),T(10,10),U(10,10),IC
C----- INTEGER M,LL,N
DO 20 I=1,M
DO 20 J=1,N
U(I,J)=O*0
DO 10 INDEX=1,LL
U(I,J)=U(I,J)+R(INDEX,I)*T(INDEX,J)
10      CONTINUE
20      CONTINUE
RETURN
END
C===== SUBROUTINE RAPRN(NMAX,M,N,L,A,IDLIN,FMT)
C===== REAL*4 L(NMAX,N)
C===== DIVNSICK FMT(IDIM)
NU=L
DO 20 NL=1,N*L
1F(NU+GT*NL)NU=N
DO 10 I=1,M
10      WRITE(6,FMT)(A(I,J),J=NL,NU)
      NU=NU+L
20      RETURN
END

```

```

C ===== SUBROUTINE RGAIN (M,NS,NC,NDE,WR,WI,VF,GN,WL,TCB,W21,LT,CT,CI,CT,M
1HS,MT) ===== CPT13E20
C ===== IMPLICIT REAL*4 (A-H,O-Z) ===== CPT13E30
C DIMENSION WR(M),WI(N),VF(M,M),GN(NS,NS) ===== CPT13E40
C DIMENSION WL(NS,NS),TCB(M,M),W21(NS,NS),LT(NS,NS) ===== CPT13E50
C DIMENSION CI(NS),CT(NS,NS) ===== CPT13E60
C K=1 ===== CPT13E70
C NRZEV=0 ===== CPT13E80
C NCPT=0 ===== CPT13E90
C NRZEV=0 ===== CPT13E90
C IF (K.GT.M) GO TO 210 ===== CPT13E90
C CHECK FOR EIGENVALUE AT OR NEAR J-OMEGA AXIS TO INCLUDE IN E-L FIGSYS
C TURN FIRST ONE POSITIVE AND SECOND ONE NEGATIVE ===== CPT13E90
C
C----- EIGVR=CAEPS(WR(K))
C EIGVR=ABSP(WR(K))
C IF (EIGVR.GE.1.E-10) GO TO 60
C IF (WI(K).GT.0,20,40
C NRZEV=NRZEV+1
C IF (NRZEV.GT.1) GO TO 30
C WR (K)=EIGVR
C GU TO 20
C WR (K)=-EIGVR
C WRITE (6,290)
C GO TO 150
C NCPTZEV=NCPTZEV+1
C IF (NCPTZEV.GT.1) GO TO 50
C WF (K)=EIGVR
C WR (K+1)=EIGVR
C GU TO 10
C WR (K)=-EIGVR
C WR (K+1)=-EIGVR
C WRITE (6,300)
C GO TO 180
C IF (WR (K)) 140,70,70
C IF (WI(K)) 110,80,110
C CONTINUE
C----- EIGENVECTOR FOR REAL EIGENVALUE, POSITIVE ===== CPT14200
C 80 IF (NOB.EQ.0) GO TO 100 ===== CPT14210
C 90 DO 90 J=1,M ===== CPT14220
C TCB(J,KP)=VF(J,K)
C KP=KP+1 ===== CPT14230
C K=K+1 ===== CPT14240
C GO TO 1C ===== CPT14250
C----- CPT14260

```

```

C-----CONTINUE FOR COMPLEX EIGENVALUE, POSITIVE REAL PART-----CPT14270
C110 IF (NCB.EQ.0) GO TO 130
C DO 120 J=1,N
C FR=VF(J,K)
C FI=-VF(J,K+1)
C TCB(J,KF)=FR+FI
C TCB(J,KF+1)=FR-FI
C KP=KP+2
C K=K+2
C GO TO 10
C IF (WI(K).EQ.0) GO TO 180
C-----EIGENVECTOR FOR REAL EIGENVALUE, NEGATIVE REAL PART-----CPT14280
C140 C(KN)=AP(K)
C1(KN)=N1(K)
C1(KN)=NE
C IF (NCB.NE.0) GO TO 170
C KN=KN+NS
C DO 160 J=1,N
C TCB(J,KNS)=VF(J,K)
C KN=KN+1
C K=K+1
C GO TO 10
C-----EIGENVECTOR FOR COMPLEX EIGENVALUE, NEGATIVE REAL PART-----CPT14290
C180 RR=WR(K)
C RI=WI(K)
C (KN)=RR
C (KN+1)=RR
C1(KN)=RI
C1((KN+1))=-RI
C IF (NCB.NE.0) GO TO 200
C KN=KN+NS
C DO 190 J=1,N
C FR=VF(J,K)
C FI=-VF(J,K+1)
C TCB(J,KNS)=FR+FI
C TCB(J,KNS+1)=FR-FI
C KN=KN+2
C K=K+2
C GO TO 10
C-----CONTINUE-----CPT14300
C190 IF (NCB.NE.0) GO TO 240
C-----FORMATION OF W11-----CPT14310
C200 DO 220 I=1,NS
C DO 220 J=1,NS
C WI(I,J)=TCB(I,J+NS)
C CT(I,J)=W11(I,J)
C-----FORMATION OF W21-----CPT14320
C DO 230 I=1,NS
C-----CONTINUE-----CPT14330

```

```

DO 230 J=1,NS
W21(I,J)=TCB(I+NS,J+NS)
CONTINUE
IF (NDE.EQ.0) GO TO 200
DO 250 I=1,NS
DO 250 J=1,NS
W21(I,J)=TCB(I,J)
W11(I,J)=TCB(I+NS,J)
CONTINUE
NSC=NS*NS
CALL MINV (NSQ, M11, NS, DETC, LT, MT)
-- CALCULATE THE REGAIN MATRIX --
DO 270 IL=1,NS
DO 270 JL=1,NS
GN (IL,JL)=0.0D0
DO 270 KL=1,NS
GN (IL,JL)=GN (IL,KL)+W21(IL,KL)*W11(KL,JL)
IF (NDE.EQ.0) RETURN
DO 280 J=1,NS
DO 280 J=1,NS
CT (I,J)=W11(J,I)
RETURN
FORMAT (1X,5H EULER-LAGRANGE EQUATIONS HAVE A REAL EIGENVALUE AT,
114H NEAR ZERO.)/
300 FORMAT (1X,4H EULER-LAGRANGE EQUATIONS HAVE A COMPLEX PAIR OF ,40
THEI EigenVALUES AT OR NEAR THE J-CMEGA AXIS.)
END
C=====SUBROUTINE MINV (NSQ, A, N, D, L, M)
C=====IMPLICIT REAL*4 (A-H,G-Z)
C=====DOUBLE PRECISION A (NSQ), L (N), D, B1GA, HOLD
C=====NM=N*N
C=====D=1.0D0
C=====D=1.0E0
NK=-N
DO 180 K=1,N
NK=NK+1
L(K)=K
M(K)=K
KK=NK+K
B1GA=A(KK)
DO 20 J=K,N
IZ=N*(J-1)
CPT14740
CPT14750
CPT14760
CPT14770
CPT14780
CPT14790
CPT14800
CPT14810
CPT14820
CPT14830
CPT14840
CPT14850
CPT14860
CPT14870
CPT14880
CPT14890
CPT14900
CPT14910
CPT14920
CPT14930
CPT14940
CPT14950
CPT14960
CPT14970
CPT14980
CPT14990
CPT15000
CPT15100
CPT15110
CPT15120
CPT15130
CPT15140
CPT15150
CPT15160
CPT15170

```

```

DO 20 I=K,N
IJK=IZ+1
IF (DABS(BIGA)-DABS(A(IJ))) .LT. 1.0,20,20
C
IF (DABS(BIGA)-DABS(A(IJ))) .GT. 1.0,20,20
BIGA=A(IJ)
L(K)=I
M(K)=J
CONTINUE
C
      IF (I-K) 50,50,30
      K1=K-N
      D5 40 I=1,N
      K1=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
      A(JI)=HOLD
      C
      50 I=M(K)
      IF (I-K) 80,80,60
      JP=N*(I-1)
      DO 70 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
      A(JI)=HOLD
      C
      60 IF (BIGA) 100,90,100
      D=0.0D0
      D=C*OE
      RETURN
C90
      100 DO 120 I=1,N
      IF (I-K) 110,110,110
      IK=NK+I
      AI(K)=A(IK)/(-BIGA)
      COUNTNE
      110
      120
      C
      DO 150 I=1,N
      IN=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 150 J=1,N
      IJ=IJ+N
      IF (I-K) 130,130,130
      IF (J-K) 140,140,140
      130

```


INTEGER IERR
DOUBLE PRECISION A
DIMENSION A(M,N)

WRITE(5,10) IERR
CALL RAFRNT(N,N,S,A,4,*(961X,1PD15.0))*

10 RETURN (25H FAILURE IN HQR2 ON EIGENVALUE NO. ,13)
END

C== SUBROUTINE MATPRT -- DISPLAYS A TWO-DIMENSIONAL ARRAY (16 Cols. MAX)=
C== IN VARIABLE SCREEN FORMAT FOR USER EASE IN ROW IDENTIFICATION.
C== SUBROUTINE MATPRT (PRT,NRCW,NCCL)

C== IMPLICIT REAL*4 (A-H,O-Z)
DIMENSION PRT(NRCW,NCCL)

C IF (NCCL .EQ. 0) NCCL = 1
IF (NCCL .EQ. 1) WRITE(5,10)
IF (NCCL .EQ. 2) WRITE(5,20)
IF (NCCL .EQ. 3) WRITE(5,30)
IF (NCCL .EQ. 4) WRITE(5,40)
IF (NCCL .EQ. 5) WRITE(5,50)
IF (NCCL .EQ. 6) WRITE(5,60)
IF (NCCL .EQ. 7) WRITE(5,70)
IF (NCCL .EQ. 8) WRITE(5,80)
IF (NCCL .EQ. 9) WRITE(5,90)
IF (NCCL .EQ. 10) WRITE(5,100)
IF (NCCL .EQ. 11) WRITE(5,110)
IF (NCCL .EQ. 12) WRITE(5,120)
IF (NCCL .EQ. 13) WRITE(5,130)
IF (NCCL .EQ. 14) WRITE(5,140)
IF (NCCL .EQ. 15) WRITE(5,150)
IF (NCCL .EQ. 16) WRITE(5,160)
RETURN

C 10 FORMAT (F12.5)
20 FORMAT ((2F12.5))
30 FORMAT ((3F12.5))
40 FORMAT ((4F12.5))
50 FORMAT ((5F12.5))
60 FORMAT ((6F12.5))
70 FORMAT ((6F12.5))
80 FORMAT ((6F12.5))
90 FORMAT ((6F12.5))
100 FORMAT ((6F12.5))
110 FORMAT ((6F12.5))
120 FORMAT ((6F12.5))

```

120      FORMAT (6F12.5,' ',6F12.5,' ',6F12.5,' ')
140      FORMAT (6F12.5,' ',6F12.5,' ',2F12.5,' ')
150      FORMAT (6F12.5,' ',6F12.5,' ',3F12.5,' ')
160      FORMAT (6F12.5,' ',6F12.5,' ',4F12.5,' ')
END
C=====
SUBROUTINE MODF (WNORM,G,GNORM,NS,N1,N2,ICON)
C
C      TRANSFORMATION MATRIX U OR U-INV
C      NO. OF STATE
C      NO. OF INPUTS OR OUTPUTS
C      CONTROL FLAG TO INDICATE WHICH TRANSFORMATION
C
C      1 = MODAL H
C      2 = MODAL C
C      3 = MODAL K
C      4 = EIGENVECTORS MATRIX X
C      5 = MEASUREMENT EIGENVECTORS MATRIX X
C      6 = IMPLICIT REAL*4 (A-H-G-Z)
C      7 = NORM(S,I,J)
DO 10 I=1,NS
    GNP(M(I,J))=C(I,J)
10   IPCINT=ICUN+1
    GU TO (4,20,90,90,90), 1PUNIT
    DO 30 J=1,NS
        DO 30 K=1,NS
            GNL(M(I,J)+WNORM(I,K)*G(K,J))
            GO TO (4,7C,90,90,80), IPCINT
40   WRITE (6,17C)
    DO 60 I=1,NS
        WRITE (6,230) (GNORM(I,J), J=1, N2)
    RETURN
70   WRITE (6,180)
    GU TO 5C
    WR ITE (6,240)
    DO 100 K=1,NS
        DO 100 J=1, NS
            DO 100 I=1, N1
                GNCRM(I,J)=GNORM(I,J)+G(I,K)*WNORM(K,J)
            GO TO (110,110,110,130,140), IPCINT
100  GNCRM(I,J)=GNORM(I,J)+G(I,K)*WNORM(K,J)
    DO 100 I=1, N1
        WR ITE (6,190)
        GO TO 15C
    DO 120 I=1, N2
        WR ITE (6,200)

```



```

      DO 50 K=1,NS
     1 IF ((K*K*EC**1) .GT. 40
          IF (DABS(WY(K)).LT.1.D-10) GO TO 50
          IF (ABS(WY(K)).LT.1.E-10) GO TO 50
     2 LC=LCT+1
     3 EMAX=C*EO
     4 CMOD=YEC*(EMAX)**2+YEC*(L,K+1)**2
     5 EMAX=C*CMD
     6 CONTINUE
     7 VMR=YEC(M,K)
     8 VM1=YEC(I,K+1)
     9 DU30=I=1,NS
    10 VR=YEC(I,K)
    11 V1=YEC(I,K+1)
    12 VECRN=(VR#VMR+VI*VM1)/EMAX
    13 VECIN=(VI-VR*VM1+VI*VMR)/EMAX
    14 VECRM(I,K)=VECRN
    15 VECRM(I,K+1)=VECIN
    16 CONTINUE
    17 KK=1 TO 50
    18 GO TO 50
    19 KK=0
    20 CONTINUE--NORMALIZE REAL EIGENVECTORS BY THE TOTAL LENGTH-----
    21 DO 80 K=1,NS
    22 IF (DABS(WY(K)).GE.1.D-10) GO TO 80
    23 IF (ABS(WY(K)).GE.1.E-10) GO TO 80
    24 LR=LRT+1
    25 REMUD=C*EO
    26 DO 60 I=1,NS
    27 REMCD=YEC(I,K)**2+REMUD
    28 RMOD=DSQRT(REMUD)
    29 RACD=DSQRT(REMUD)
    30 DO 70 I=1,NS
    31 RVEC=YEC(I,K)/RMOD
    32 VECRM(I,K)=RVEC
    33 CONTINUE
    34 DO 50 I=1,NS
    35 REMCD=DSQRT(REMUD)
    36 RMOD=DSQRT(REMUD)
    37 DO 60 I=1,NS
    38 RVEC=YEC(I,K)/RMOD
    39 VECRM(I,K)=RVEC
    40 CONTINUE
    41 GLTO(SC100,110,120,130), IWRITE
    42 90 WRITE(1,320)
    43 GO TO 140
    44 WRITE(1,330)
    45 GO TO 140
    46 WRITE(1,340)
    47 GO TO 140
    48 C
    49

```

```

120 WRITE (6,350)
C GO TO 140
130 WRITE (6,360)
140 KK=0
NPRTW=C
NFMTH=1
DD 180 I=1,NS
IF (KK.EQ.1) GO TO 170
IF (ABS(WY(I)).GT.I-10) KK=1
PRINT NOT MORE THAN 5 OR
IF (NPRTW+1)=RIGHT
FMT (NFMTH+1)=RIGHT
WRITE (6,FMT) (STORE(J),J=1,NPRTW)
NPRTW=C
NFMTH=1
NPRTW=NPRTW+1
NFMTH=NPRTW+1
IF (KK.EQ.1) GO TO 160
STORE(NFMTH)=FIELD
NFMTH=NFMTH+1
FMT(NFMTH)=SEMCOL
GO TO 180
STORE(NPRTW)=WZ(1)
FMT(NFMTH)=FIELD
FMT(NFMTH+1)=COMMA
STORE(NPRTW+1)=WY(1)
FMT(NFMTH+2)=FIELD
FMT(NFMTH+3)=SEMCOL
NFMTH=NPRTW+1
NPRTW=NPRTW+1
GO TO 180
KK=0
CONTINUE
FMT(NFMTH)=SEMCOL
FMT(NFMTH+1)=RIGHT
WRITE (6,FMT) (STORE(J),J=1,NPRTW)
IF (IWRITE.NE.1) GO TO 190
WRITE (6,370)
GO TO 200
CONTINUE
WRITE (6,380)
170
180
CONTINUE
FMT(NFMTH)=SEMCOL
FMT(NFMTH+1)=RIGHT
WRITE (6,FMT) (STORE(J),J=1,NPRTW)
IF (IWRITE.NE.1) GO TO 190
WRITE (6,370)
GO TO 200
CONTINUE
C 190
C 200
CALL RAPPNT (NS,NS,6,WRITM4,6,(1X,1PE13.6))
CALL RAPPNT (210,220,220,1,WRITE
CALL MCCE (NORM,HG,CW,NS,NI,NZ,5)
C 210
GO TO 220

```

164

CALL WURZEL(A,B,N,THETA,EPS,NC)

DESCRIPTION OF PARAMETERS
A - SYMMETRIC, POSITIVE-DEFINITE, INPUT SQUARE ROOT OF A, AFTER EXIT
B - UNKNOWN, CONTAINS SQUARE ROOT OF A.
N - ORDER OF A AND B. (SEE METHOD). IF EIGENVALUES
THETA - CONSTANT (SEE METHOD) USE THE CLOSEST ONE TO GUESS VALUE.
OF A ARE UNKNOWN WHICH DESIRED ACCURACY (SEE METHOD).
EPS - CONSTANT SPECIFYING ACCURACY CHOOSE EPS = 1.E-6.
NC - FOR SIX PLACES OF A AND B.
- ORDER OF A AND B.

REMARKS
PARAMETERS A, B, THETA, AND EPS ABOVE ARE ALL REAL*8

THE CONVERGENCE PROCESS IS VERY SLOW IF THE LARGEST AND
SMALLEST EIGENVALUES OF A DIFFER BY SEVERAL ORDERS OF
MAGNITUDE.

SUBROUTINES AND FUNCTION SUBPROGRAM REQUIRED
NONE

METHOD
IF X IS THE VECTOR ASSOCIATED WITH MATRIX A, THEN MATRIX A
IS POSITIVE-DEFINITE IF X*A*X IS GREATER THAN ZERO.
IN THE SEQUENCE:
$$B(K+1) = B(K) + C * (A - B(K) * C) * C$$
 ...1..
CONVERGENCE TO THE SQUARE ROOT OF A.
IN EXPRESSION CONSEQUENTLY C = THETA/2*SQRT(NORM(A)), WHERE:
THE SQUARE ROOT OF ALPHA IS THE QUOTIENT OF NORM(A) DIVIDED BY THE MAXIMUM
EIGENVALUES OF A, AND IT IS GREATER OR EQUAL TO ONE.
NORM(A) IS THE MAXIMUM VALUE OBTAINED BY ADDING EACH ROW
OF A ALL THE WAY ACROSS. THE FIRST APPROXIMATION OF B, THE SQUARE ROOT MATRIX, IS
OBTAINED AS FOLLOWS: B(0) = 2.C*C*A

THE RATE OF CONVERGENCE OF CONVERGENCE TO THE SQUARE ROOT OF A
IS GIVEN BY THE RATE OF CONVERGENCE TO ZERO OF THE
SEQUENCE: $X^{(K)} = 1 - \Theta(\text{SQRT}(\text{LAMAMIN/LAMAX}))^{**K}$
WHERE LAMAMIN AND LAMAX ARE THE MINIMUM AND MAXIMUM
EIGENVALUES OF A, RESPECTIVELY.

TO ACCELERATE CONVERGENCE THETA SHOULD BE CHOSEN CLOSE TO
THE SQUARE ROOT OF ALPHA

IF NOTHING IS KNOWN ABOUT ALPHA, OPTIMUM CHOICE OF THETA IS
A VALUE CLOSE TO ONE.

THE ITERATION FOR COMPUTATION OF SQRT(A) STOPS WHEN:
MAX(ABS(B(I,J)**(K+1) - B(I,J)**K)) LESS THAN EPS
I, J

SEE "COMMUNICATIONS OF THE ACM", VOLUME 10, NUMBER 3, MARCH
1967, ALGORITHMS., ALGORITHM 298, PAGE 182

SUBROUTINE JRZEL(A,B,N,THETA,EPS,ND)

IMPLICIT REAL*4 (A-H,O-Z)
DIMENSION A(ND,ND),B(ND,ND),BB(100)

FORMAT(•C•/(4E17.9),•E17.9)

1000 FORMAT(•DELTAB=•E17.9)
PART1. DETERMINATION OF C,

C=0

DO 20 I=1,N

S=0

DO 10 J=1,N

1C S=S+A(I,J)

IF (S-C) 2C,20,15

15 C=S

2C CONTINUE

C=C*5*THETA/ SQRT(C)

SET B(C)

3C DO 40 I=1,N

DO 40 J=1,N

E(I,J)=2.0*C*A(I,J)

4C B(J,I)=E(I,J)

C PART2. ITERATION FOR COMPUTATION OF SQRT(A) ITERATION STOPS WHEN:

MAX(ABS(B(I,J)**(K+1) - B(I,J)**K)) LESS THAN EPS

5C DELTA = C.

DO 80 I=1,N

DO 70 J=1,N

S=0

DO 60 K=1,N

S=S-B(I,K)*B(K,J)

6C BB(J)=B(I,J)+C*(A(I,J)+S)

7C

C COMPUTE MAXIMUM VALUE AS SHOWN IN PREVIOUS COMMENT.

```
      DO 75 J=1,N
      S= ABS(E(I,J))-BB(J)
      IF (S-DELT)75,75,74
      DELTA=S
      74 WRTE(C,100)DELTA
      75 B(I,J)=BB(J)
      80 CONTINUE
```

C SET SYMMETRIC TERMS IN MATRIX B.

```
NN=N-1
DO 90 I=1,NN
  NJ=I+1
  DO 90 J=NJ,N
    9C B(J,I)=E(I,J)
    IF (J-I)1001)((B(I,LL),LL=1,N),II=1,N)
    94 GO TO 54
    95 RETURN
    END
*** THE END ***
```

LIST OF REFERENCES

1. Kalman, R.E., "When is a Linear Control System Optimal?", J. of Basic Eng., ASME, Vol 86, pp. 51-60, March 1964.
2. Athans, M., "The Role and Use of the Stochastic Linear Quadratic Gaussian Problem in Control System Design", IEEE Tran. Automat. Control, vol AC-16, pp. 529-552, December 1971.
3. Bryson, A. E. and Y. Ho., Applied Optimal Control, Waltham, MA, Ginn, 1969.
4. Kwakernaak, H., and Sivan, R. Linear Optimal Control System, Wiley International, New York, 1972.
5. Tyler, J. S., "The Use of a Quadratic Performance Index to Design Multivariable Control System," IEEE Trans. Auto. Control, Vol 11, pp. 84-92, 1966
6. Fallside, F. and Seraji, H., "Design of Optimal Systems by a Frequency Domain Technique," Proc. IEE, Vol 117, pp. 2017-2024, 1970.
7. MacFarlane, A. G. J., "Return Difference and Return Ratio Matrices and Their Use in Analysis and Design of Multivariable Feedback System," Proc. IEE, Vol 117, pp. 2037-2049, 1970.
8. Solheim, O. A. "Design of Optimal Control Systems with Prescribed Eigenvalues," Int. J. Controls, Vol. 15, No. 1, pp. 143-160, 1972.
9. Harvey, C. A. and Stein, G., "Quadratic Weights for Asymptotic Regulator Properties," IEEE tran. Auto. Control, Vol. AC-23, June 1978.
10. Stein, G. "Generalized Quadratic Weights for Asymptotic Regulator Properties," IEEE Tran. Auto. Control, Vol AC-24, pp. 559-567, August 1979.
11. Broussard, J. R., "A Quadratic Weight Selection Algorithm," Proc. of the 1981 Joint Auto. Contr. Conference, 1981.
12. Gully, S. W., Sandell, N. R. and Lee, W. H., "A Robustness Synthesis Method," IEEE Decision and Control Proc. 21, 1982.

13. Dolye, J. C., and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," IEEE Trans. Auto. Control, Vol. 26, No. 1 PP. 4-16, Feb 1981.
14. Safanov, M. G. and Athans, M. "Gain and Phase Margin of Multiloop LQG Regulars," IEEE Trans. Auto. Control, April 1977.
15. Lehtomaki, N.A., N.R. Sandell, Jr. and M. Athans, "Robustness Results in Linear Quadratic Gaussian Based Multivariable Control Designs," IEEE Trans. Auto. Contr., Vol. AC-26, No.1, Feb. 1981
16. Moore, B. C., "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment," IEEE Tran. Auto. Contr., AC-20, pp. 689-692, October 1976.
17. Dooren, P. V., Kautsky, J. and Nichols, N. K., "Robust Pole Assignment in Linear State Feedback," Int. J Contr. Vol 41. No 5. pp. 1129-1155, 1985.
18. Kwakernaak, H., Asymptotic Root Loci for Multivariable Linear Optimal Regulator, IEEE Tran. Auto. Contr., AC-21, pp. 378-381 1975.
19. NASA TM-84524, Application of Matrix Singular Value Properties for Evaluating Gain and Phase Margins of Multiloop Systems, Mukhopadhyay, V. and Newsom, J. R., July 1982.
20. Vanderplatts, G. N., ADS - A Fortran Program for Automated Design Synthesis, - Version 1.0, Naval Postgraduate School, Monterey, California, May 1984.
21. Vanderplatts, G. N., Numerical Optimization Techniques for Engineering Design: With Applications, McGraw Hill, 1984.
22. Gordon, V. C., Utilization of Numerical Optimization Techniques in the Design of Robust Multi-input Multi-output Control Systems, Ph.D. Dissertation, Naval Postgraduate School, Monterey, California 1984.
23. Alpatech Inc. TR-121, Multivariable Stability Margins for Vehicle Flight Control System, Sandell, N. R., Jr., et al, Dec. 1981
24. NASA Report 3889, Design of Multivariable Feedback Control Systems Via Spectral Assignment Using Reduced-Order Models and Reduced-Order Observers, R. R. Mielke, L. J. Tung and P. I. Carraway III, April 1985

INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145		2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100		2
3. Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5100		1
4. Prof. D. J. Collins, Code 67Co Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5100		2
5. Mr. Chow Wah Keh #08-5067 Blk 639, Ang Mo Kio Ave. 6 S'pore 2056 Republic of Singapore		3

Thes Thesis
C448 C448878

c.1 c.1

Chow
Automated pole
placement algorithm
for multivariable
optimal control
synthesis.

3 NOV 88

32404

216193
Thesis
C448878 Chow
c. 1 Automated pole
placement algorithm
for multivariable
optimal control
synthesis.



thesC448878
Automated pole placement algorithm for m



3 2768 000 64784 6
DUDLEY KNOX LIBRARY